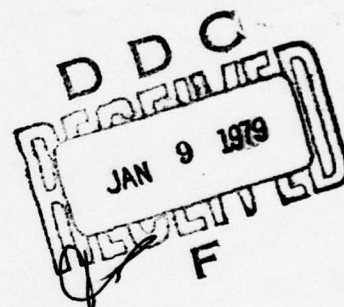LEVEL II

C

A Numerical Method for Shape-From-Shading
from a Single Image

2LT Thomas M. Strat
HODA, MILPERCEN (DAPC-OPP-E)
200 Stovall Street
Alexandria, VA 22332

Final Report
19 January 1979

79 01 08 046

A Numerical Method for Shape-From-Shading From a Single Image.

by
Thomas M. Strat

19 Jan 79

113 p.

ABSTRACT

The shape of an object can be determined from the shading in a single image by solving a first-order, non-linear partial differential equation. The method of characteristics can be used to do this, but it suffers from a number of theoretical difficulties and implementation problems. This thesis presents an iterative relaxation algorithm for solving this equation on a grid of points. Here, repeated local computations eventually lead to a global solution.

The algorithm solves for the surface orientation at each point by employing an iterative relaxation scheme. The constraint of surface smoothness is achieved while simultaneously satisfying the constraints imposed by the equation of image illumination. The algorithm has the distinct advantage of being capable of handling any reflectance function whether analytically or empirically specified.

Included are brief overviews of some of the more important shape-from-shading algorithms in existence and a list of potential applications of this iterative approach to several image domains including scanning electron microscopy, remote sensing of topography and industrial inspection.

Thesis Supervisor: Berthold K. P. Horn

Title: Associate Professor of Electrical Engineering and Computer Science

79 01 08 046

TABLE OF CONTENTS

List of Figures

1. INTRODUCTION

Making machines more useful is a major goal of artificial intelligence. One obvious way of making machines more useful is to enable them to deal directly with their environment. Making machines "see" is one way to do this. How to make machines see is not so obvious.

Simply put, the goal of machine vision is to develop systems which take an image, whether it be a photograph, an X-ray or a painting, and have it produce a symbolic description of the object(s) within the image. The design of such a system is a matter of great debate, as is the form of the description itself.

In this thesis, we are concerned with one small part of this difficult transformation from image to description -- that of computing the shape of a smooth surface from an image of that surface.

1.1 What is Image Analysis?

The purpose of machine vision is to define and describe the components of a scene given an image of that scene. Historically, this process has been divided into two parts. The purpose of *image analysis* is to extract features from a raw image and to convert those features into a convenient symbolic representation. The purpose of *scene analysis* is to interpret the symbolic features produced by image analysis according to some externally defined goal.

Early artificial intelligence research in machine vision concentrated on images of scenes containing plane-faced polyhedra. Initially, the distinction between image analysis

and scene analysis seemed quite clear. The purpose of image analysis was to generate a two-dimensional line drawing of the scene [Horn, 1973]. The purpose of scene analysis was to interpret a two-dimensional line drawing in terms of the three-dimensional objects which gave rise to it [Roberts, 1965; Guzman, 1968; Huffman, 1971; Clowes, 1971; Turner, 1971; Mackworth, 1973; Waltz, 1975; Winston, 1975]. As the field matured, the actual distinction between image analysis and scene analysis became less clear. More recent work [Winston, 1973; Shirai, 1975; and Freuder, 1976] made use of a richer form of interaction between image analysis and scene analysis. Nevertheless, a conceptual distinction between the two still exists.

The shape-from-shading problem lies within the realm of image analysis. It deals directly with an image as input to determine a representation of shape suitable for subsequent scene analysis.

1.2 The Difficulties with Image Analysis

Despite strong motivation and years of concerted effort, researchers have failed to come up with a "universal" shape-from-shading method. To be sure, inroads have been made in many specialized areas but each approach involves many assumptions about the imaging situation and is applicable only in limited circumstances. What is it that makes the analysis of images so elusive?

The purpose of this section is to point out some of the difficulties associated with the interpretation of image intensities. Knowledge of these difficulties is necessary to predict when and why certain image analysis techniques will work and when and why they will fail.

1.2.1 The Data in an Image

A great deal of information is contained in the intensity values recorded in an image, and this massive quantity of data has proven to be a stumbling block to image analysis. Image analysts often rely on data compression and forget about actual image intensities as soon as possible. One method of image analysis is to extract features of intensity which are important and to throw away everything else, but those features one can extract easily are those which can be conveniently defined in terms of properties of images. Properties of images, however, do not usually correspond directly with properties of the objects which gave rise to them.

Practical vision systems exist for domains in which there is a direct correlation between properties of images of the domain and interesting properties of objects in the domain. Domains which are inherently two-dimensional generally provide such a good correlation. Optical character recognition [IBFI, 1969], blood cell analysis [Young, 1969], and automatic fingerprint identification [Grasselli, 1969], are three such examples.

The research in this thesis attempts to exploit *all* the data in an image rather than to compress the data into a more manageable, reduced size. The transformation from object space to an image space is a functional mapping from an object point (x, y, z) to an image point (u, v) and a corresponding intensity value I. Roughly speaking then, the dimensionalities of the two domains match. Difficulties arise from the fact that the image generating transformation is many-to-one. Therefore the inverse transformation (the solution of the shape-from-shading problem) is not uniquely determined without further

assumptions. The physical interpretation is that any number of surfaces can give rise to the same image, so shape-from-shading can only be achieved by imposing constraints in the form of prior expectations about the imaged surface.

1.2.2 Image Illumination

The image depends on more than just the shape of the surface and the location from which it is viewed. As we all know, object surfaces appear differently at different times of day. In fact, a single surface can have an infinite number of images depending upon the distribution of incident illumination. Changes in illumination can cause a surface to appear quite differently even when viewed from the same direction.

1.2.3 Surface Photometry

A third factor confounding the shape-from-shading problem is the fact that different surfaces reflect light in different ways. The composition of the surface of an object determines how much light is reflected and in what directions. As a result, identically shaped surfaces under identical lighting conditions can give rise to different images. Even objects of the same material appear differently depending upon whether they are wet, dry, clean or dirty. The conclusion is that objects of the same shape under identical lighting conditions can give rise to different images.

1.2.4 Human Performance

Humans are remarkably successful at interpreting image intensities despite the problems caused by projection, illumination and surface photometry. The fact that humans are capable of interpreting single images of arbitrarily shaped, unfamiliar objects rules out any need for high-level information and any need for more than one image. It seems that the determination of shape from the shading information in an image must be possible since the human visual system can achieve it.

The numerical approach that is presented in this paper for solving the shape-from-shading problem is not intended to reflect the way the human (or any other animal's) visual system works. The desire is the less ambitious, yet useful, goal of designing a mechanical system capable of determining shape from a single image.

1.3 The Problem

As we have seen, a generally applicable shape-from-shading machine must deal with a wide variety of difficult problems. The differences among present algorithms can be viewed in terms of which complications are actually solved and which are avoided entirely (by simplifying assumptions). For example, occluding contours pose a problem to some shape-from-shading techniques whereas restricting the domain to images of smooth surfaces containing no occluding contours is a way to avoid this problem.

This thesis presents a practical shape-from-shading algorithm which sidesteps some of these complications to be sure, but also overcomes some previously insurmountable ones. The method which will be described is capable of ascertaining surface shape from the

shading information in a single image. The only information required beside the image itself is the reflectance function of the surface and some suitable set of initial conditions, provided the surface is smooth at all points in the region to be analyzed.

Heretofore, only the analytic approach due to Horn [1975], of solving a first-order, non-linear, partial differential equation was capable of determining surface shape from a single image. However, Horn's method is practical only when the reflectance can be described as a simple analytic function of the surface gradients, since it requires the derivatives of the reflectance map. The algorithm of this thesis is capable of determining surface shape for nearly any reflectance function. In fact, the reflectance need not be known analytically; an empirically defined reflectance function works just as well. Both algorithms are restricted to image regions of smooth surfaces with known photometric properties.

The algorithm is posed as an iterative relaxation scheme. It seeks to simultaneously satisfy the constraints of the equations of image formation and surface smoothness at all points in the image. Global constraint is achieved by propagating pseudo-local smoothness operators throughout the image. The goal is convergence to the unique surface shape that gave rise to the image.

It is important to point out that the numerical shape-from-shading algorithm is not intended to be a stand-alone system. Rather, it performs one small part of the transformation from image to high level description. It is up to other methods to isolate regions of smooth isotropic surfaces. Then this algorithm can be utilized to determine the shape within those regions.

1.4 Applications for Shape-From-Shading

The ultimate shape-from-shading algorithm would be capable of determining the shapes of all visible surfaces in an arbitrary scene. However, as we have seen, many assumptions about a domain must be made to keep things tractable. Several domains which possess properties that can be exploited for image analysis are described in this section.

Planetary Mapping:

Images returned from satellites provide one worthwhile domain for image analysis. Images of the moon from the Apollo missions, of Mars from the Viking spacecraft, and of the earth from LANDSAT are examples of likely candidates. In theory, a shape-from-shading machine could determine the surface topography of a portion of a planet from a single satellite photo. The absence of complicating features such as cities, clouds, and variations in surface vegetation in images of the moon, Mars and Mercury provide a major simplification that renders them suitable for analysis.

The Bin of Parts Problem:

Automation of assembly lines in factories often requires knowledge of the spatial position and orientation of a part. This knowledge is especially difficult to acquire when the parts lie in a pile or in various orientations on a conveyor belt. Machine vision can bridge the gap. Properties of the intensities recorded in an image of a part can be directly related to the position and orientation of the part. Identification of a particular object in an image of many different objects is also possible.

The Scanning Electron Microscope:

The scanning electron microscope (SEM) produces images which are particularly easy to interpret because the intensity recorded is a function of the orientation of the object at that point and thus gives rise to a form of shading. This differs from the situation in optical and transmission electron microscopes where intensities depend instead on the thickness and optical or electron density at each point. The geometry of the scanning electron microscope allows several simplifications in the algorithm for determining shape from shading. Additionally, it should be easy to combine the SEM with a minicomputer to obtain three-dimensional information because of the

random access capability of the microscope beam [Horn, 1975]. A shape-from-shading algorithm for SEM images would be especially useful because, at the magnifications used, no other way exists to accurately determine the three-dimensional shape.

Automatic Visual Inspection:

Many tasks of inspection involve the routine search for particular features in an image of an object. The repetitive nature of these tasks makes it desirable to accomplish this automatically. The fact that all images to be analyzed may be of the same object under similar lighting conditions allows one to ignore the effects of lighting and utilize the properties of the object to be imaged. Inspection of defects in metal castings [Woodham, 1978a] and military surveillance are two leading examples.

2. THE TOOLS

In the course of previous research, several mathematical and theoretical formalisms have been developed for use in image analysis. For the uninitiated, this chapter describes those tools which facilitate discussion of the concepts presented in the thesis.

2.1 Definitions

To prevent confusion between terminology used here and that from other disciplines, this section defines many of the relevant terms which may otherwise be somewhat ambiguous.

For our purposes, an *image* is any function of two variables which could have been generated by the procedures described in the remainder of this chapter. That is, an image is nothing more than our intuitive notion of a shaded picture. Digital computers sometimes require a *digitized image*, which is simply a set of intensity values corresponding to a finite number of image points usually selected to lie on a square grid. A *synthetic image* is actually a digitized image that has been produced mathematically by a digital computer in a way that models the normal imaging process. A square grid is often superimposed upon a real image to select an *image point* (u, v) at each vertex. The *neighbors* of image point (u, v) are those image points which are closest to (u, v). The size of a *neighborhood* depends on the context in which the term is used.

For reasons discussed later, each image point has an associated *surface point* on the *object* which gave rise to that image. Every surface point has a unique *local surface*

orientation which is the orientation of a plane tangent to the surface at that point. The solution of the shape-from-shading problem will be in the form of the local surface orientation at the surface point associated with every image point. Often we will speak of the "orientation at an image point" which is to be taken as an abbreviation for the local surface orientation of the surface point associated with that image point. The *shape* or *topography* of a surface will be represented by the local surface orientation at a set of surface points. One may recover explicit depth values by integrating the local surface gradients over the entire region, so information in this representation is essentially equivalent to explicit knowledge of depth values of surface points to within a constant of integration. A surface with continuous first partial derivatives is called *smooth*.

Several quantities associated with illumination and the reflection of light need to be defined as well. *Irradiance* is the density of the incident flux while *radiance* is the flux emitted per unit surface area per unit projected solid angle [Nicodemus, Richmond and Hisa, 1977]. Image irradiance is often referred to as *image intensity*. *Grey-levels* are quantized measurements of image irradiance. Objects which have the same photometric properties at all surface points are referred to as *isotropic*.

2.2 Image Generation

To understand the formation of an image, one must consider two separate processes. One deals with the geometry of projection while the other deals with the intensity of light recorded in an image. Thus the generation of a synthetic image consists of determining where in an image to place a surface point and what to record at that image

point [Strat, 1978].

2.2.1 The Transformation from Object Space to Image Space.

In order to calculate the image point (u, v) which corresponds to a particular surface point (x, y, z), we can consider the projection of that surface point onto the image plane as shown in Figure 1. (To avoid inverting the image, it is convenient to think of the image plane as in front of the lens rather than behind it.) For simplicity, the lens (the viewpoint) is positioned at the origin and the image plane is perpendicular to the Z axis. In Figure 1, f is the focal length (the distance between the viewpoint and the image plane). As can be seen, a straight line connects the viewpoint, the image point and the surface point. By the proportionality of similar triangles,

$$u / f = x / z \quad \text{and} \quad v / f = y / z \tag{2.1}$$

so

$$u = \frac{f}{z} x \quad \text{and} \quad v = \frac{f}{z} y \tag{2.2}$$

These equations, which determine an image point (u, v) corresponding to object point (x, y, z), define the standard *perspective projection*. If the size of the objects in view is small compared to the viewing distance, then for all surface points (x, y, z), z is nearly constant and Equations (2.2) become (after scaling the image by the constant z/f):

$$u = x \quad \text{and} \quad v = y \tag{2.3}$$

which define the standard *orthographic projection*. The projection of images obtained using a telephoto lens is approximately orthographic. With the assumption of orthographic projection, all rays from the surface to the image plane are parallel, so the use of separate
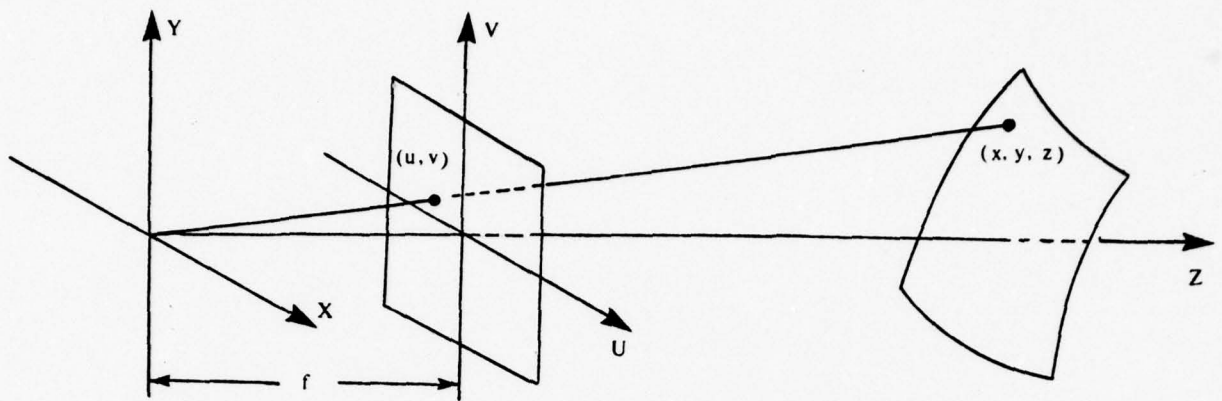
(a) Perspective Projection
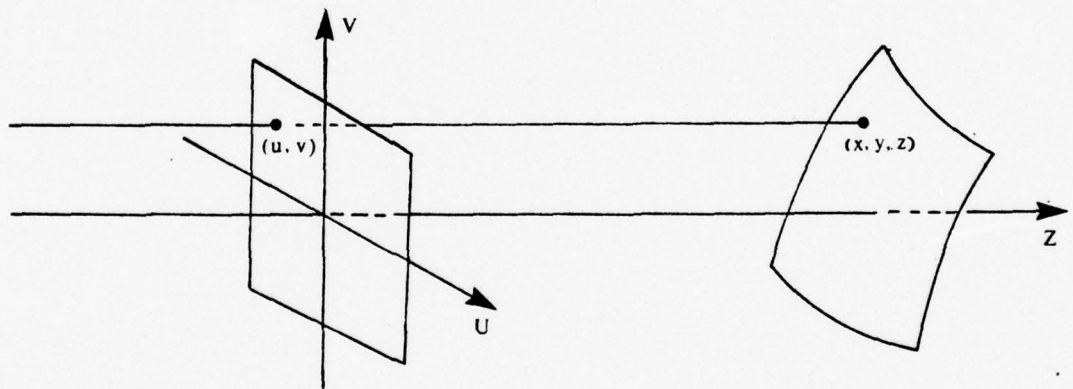
(b) Orthographic Projection

<u>Figure 1 Geometry of Image Projection</u>
The geometry of perspective projection is given in (a). A straight line connects the viewpoint, the image point and the surface point. The focal length, f, is the distance between the viewpoint and the image plane. When the viewpoint is far compared to the object's size, the lines connecting image points to object points become parallel. This projection is orthographic as shown in (b).

image coordinates is redundant, and image coordinates (x, y) and object coordinates (x, y) can be referred to interchangeably.

A word of caution is in order here. Because the projection (orthographic or perspective) is from three dimensions to two dimensions, some information is lost. It is possible that more than one object point be projected into the same image point. Because our visual world usually consists of opaque objects, only the point that is nearest the viewer will generally be visible. That is, of all the object points (x_i, y_i, z_i) that project into image point (u,v), only the one with the smallest z_i will appear at (u, v) in the image. All others will not appear in the image. The implication for the inverse projection is as follows. Assume image point (u, v) has been found to correspond to object point (x_o, y_o, z_o). Then no object points occur along the line connecting image point and object point for which $z < z_o$.

A corollary of this hidden surface phenomenon is the presence of occluding contours. Two points which are adjacent in the image do not necessarily correspond to two points which are adjacent on the object, even if the object has a smooth surface. One part of an object can obscure another.

In the work that follows, orthographic projections and images of smooth surfaces without occluding bounds are generally assumed. Occasionally, a method will be applicable to perspective projection as well and this will be pointed out.

2.2.2 The Determination of Grey-Levels in an Image

The last section described *where* a point on the surface of an object will appear in an image of that object, given a particular imaging geometry. This section deals with *what* grey-level gets recorded at that point given the imaging geometry and the photometric properties of the object.

2.2.2.1 Imaging Geometry

When a ray of light strikes the surface of an opaque object, it may be absorbed or reflected. The intensity at a point in an image of that object will depend only on the amount of light that is *radiated* (reflected) toward the viewer.

The amount of light radiated in a particular direction by a surface element depends on the orientation of the surface and the distribution of light sources around it, as well as on the nature of the surface material. The effect of the nature of the surface is described by its *photometric properties* and depends on the surface microstructure of the object material. Naturally, what constitutes microstructure depends on one's point of view. For our purposes, surface structures not resolved in a particular imaging situation will be considered microstructure. For most surfaces, there is a unique value of radiance for a given surface orientation no matter how complex the distribution of light sources.

The simplest case is that of a single point source where the geometry of reflection is governed by the three angles shown in Figure 2. The incident angle between the local normal and the incident ray is called i, while e is the view angle between the local normal and the emitted ray and g is the angle between the incident and emitted rays and is termed

the phase angle. The fraction of incident illumination at a given surface point that is reflected in the direction of the viewer is given by the reflectance function $\phi(i, e, g)$. Cases with a more complicated distribution of light sources can be modeled simply by the superposition of single point sources.
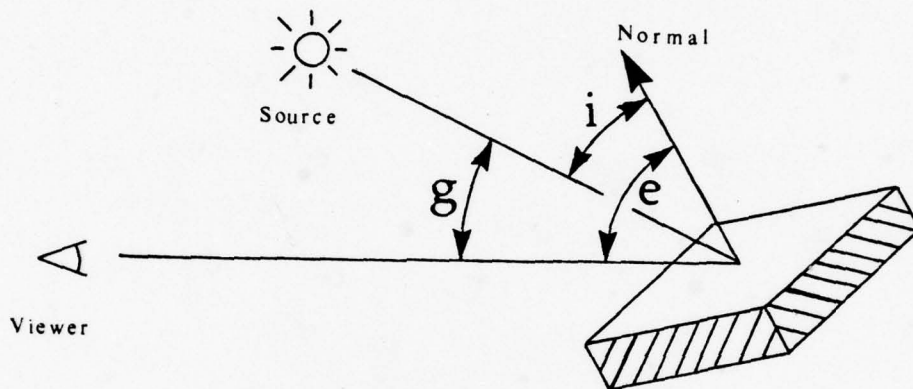
Figure 2. Geometry of Reflection This figure shows the relationship between the various angles at a particular surface element. Angles i, e and g are called the incident, emittant and phase angles respectively.

2.2.2.2 Gradient Space

It is necessary to have a convenient way to represent surface orientation explicitly. Gradient space, as popularized by Huffman [1971], and Mackworth [1973], and the "slant/tilt" formalism [Stevens, 1979] are two useful representations for reasoning about surface orientation. Because it simplifies the equations of the numerical shape-from-shading algorithm, gradient space is the only one we will pursue here.

If the equation of a smooth surface is given as $z = f(x, y)$, then the surface normal toward the viewer at the point (x, y) is

$$\left[\frac{\partial f(x, y)}{\partial x} , \quad \frac{\partial f(x, y)}{\partial y} , \quad -1 \right]$$

it is convenient to define

$$p = \frac{\partial f(x, y)}{\partial x} \qquad \text{and} \qquad q = \frac{\partial f(x, y)}{\partial y} \qquad\qquad (2.4)$$

so that the surface normal becomes (p, q, -1). The quantity (p, q) will be called the *gradient* and *gradient space* is defined to be the two-dimensional space of all such points (p, q).

We should look at some examples in order to gain a feel for gradient space. Given our viewer-centered representation, the direction to the viewer maps into the origin in gradient space. The distance from the origin in gradient space corresponds to the inclination of a plane with respect to the view vector. We find that the distance from the origin equals the slope of the surface with respect to the direction toward the viewer, *i.e.* tan(e). Additionally, the angular position of a point in gradient space corresponds to the direction of steepest descent on the object surface.

2.2.2.3 The Reflectance Map

For a given type of surface and a given distribution of light sources, there is a fixed value of radiance for every orientation of the surface normal and hence for every point (p, q) in gradient space. Thus image intensity is a single-valued function of p and q.

We need to define the relationship between the angles i, e and g and gradient point (p, q). It is convenient to work with the cosines of the angles,

$$I = \cos(i); \qquad E = \cos(e); \qquad G = \cos(g)$$

since these can be obtained easily from dot products of the three unit vectors. Suppose for

now that we have a single distant light source and that its direction is given by a vector $(p_s, q_s, -1)$. From Figure 1b it can be seen that the direction toward the viewer from any surface point is $(0, 0, -1)$ for an orthographic projection, and the surface normal is $(p, q, -1)$. So

$$G = \frac{1}{\sqrt{1+p_s^2+q_s^2}} \tag{2.5}$$

$$E = \frac{1}{\sqrt{1+p^2+q^2}} \tag{2.6}$$

$$I = \frac{1+p_s p+q_s q}{\sqrt{1+p_s^2+q_s^2}\ \sqrt{1+p^2+q^2}} \tag{2.7}$$

It is now apparent that G is constant given our assumption of orthogonal projection and distant light source. We can then derive the reflectance map $R(p, q)$ from an arbitrary photometric function $\phi(I, E, G)$ by solving the above equations for p and q in terms of I, E and G. The details are tedious and are omitted here, but the results can be found in [Horn, 1977a].

2.3 Determination of Reflectance Maps

In this section we focus on the issues of what the reflectance map might look like and how it is obtained. See [Horn, 1977a] for further details on reflectance maps.

2.3.1 Analytic Reflectance Functions

A particularly simple case is that of a *lambertian* or *matte* surface. This type of surface looks equally bright from all directions and the radiance depends only on the cosine of the incident angle.

If we consider a point source at $(p_s, q_s, -1)$ not near the viewer, the reflectance map becomes

$$R(p, q) = \cos(i) = \frac{(p_s, q_s, -1) \cdot (p, q, -1)}{\|(p_s, q_s, -1)\| \; \|(p, q, -1)\|} = \frac{1 + p_s p + q_s q}{\sqrt{1 + p_s^2 + q_s^2} \; \sqrt{1 + p^2 + q^2}} \qquad (2.8)$$

Setting R constant gives us a second-order polynomial in p and q showing that loci of constant reflectance are conic sections. The line separating lighted from self-shadowed regions, the *terminator*, is a straight line satisfying $1 + p_s p + q_s q = 0$. Similarly, the locus of $R(p, q) = 1$, the maximum value, is the single point (p_s, q_s). Contours of constant $R(p, q)$ are plotted in Figure 3 for the case $p_s = 0.7$ and $q_s = 0.3$.

White paint consists of small transparent pigment particles such as SiO_2 or TiO_2 of high refractive index and small size suspended in a transparent medium of low refractive index. This arrangement ideally reflects light equally in all directions and is an example of a real material closely approximating the ideal lambertian surface. Other examples are fresh snow, crushed glass and many flat paints.

The reflectance maps of some other surfaces have been approximated analytically [Horn, 1977a]. These include surfaces with both a matte and specular component of reflection, the material in the maria of the moon when viewed from great distances, and some substances when imaged by a scanning electron microscope.
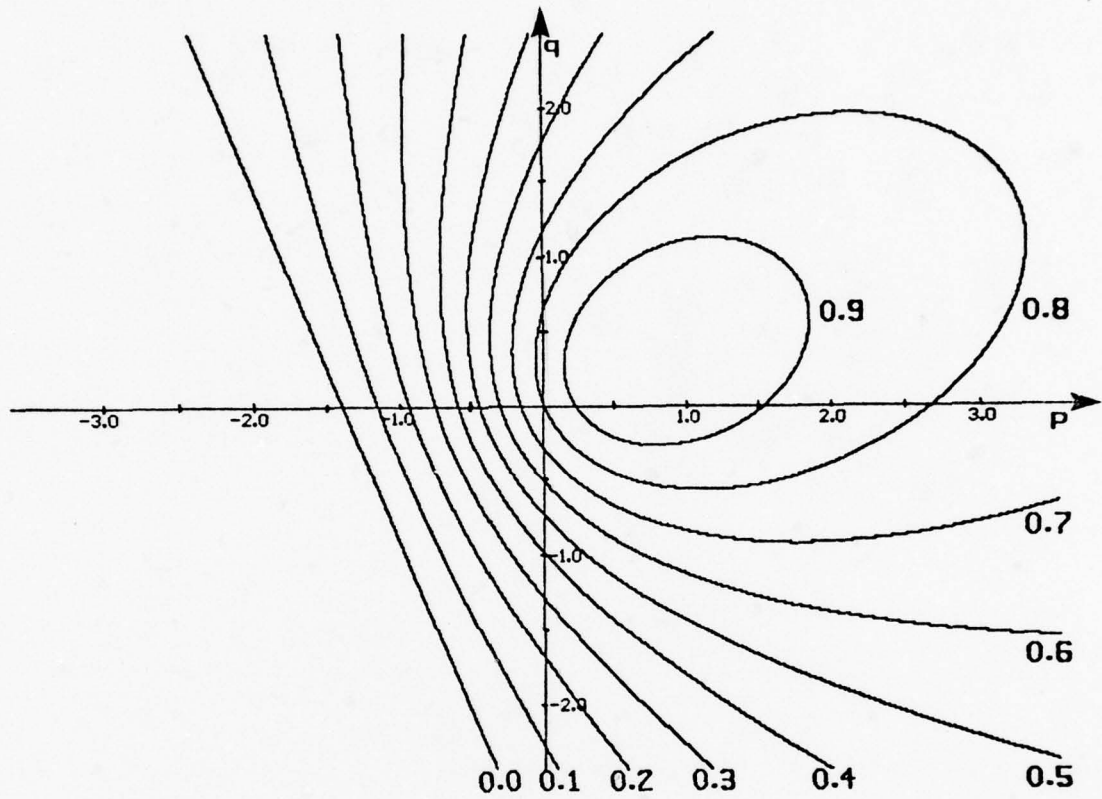
Figure 3 The Lambertian Reflectance Map
This is the reflectance map for matte surfaces when the light source is not near the viewer.
Contours of constant reflectance are shown.

$$R(p, q) = \cos(i) = \frac{1+p_s p+q_s q}{\sqrt{1+p_s^2+q_s^2} \ \sqrt{1+p^2+q^2}}$$

The direction to the single light source is $(p_s, q_s) = (0.7, 0.3)$.

2.3.2 Empirical Reflectance Functions

For most surfaces, it is not possible to determine the reflectance function in closed form. One might hope to predict reflectance functions on a theoretical basis starting with some assumed microstructure of the surface. For example, many paints can be analyzed in this manner. However, little hope exists for modeling real surfaces well enough and still being able to solve the resulting set of equations, so one must resort to experimental techniques.

One way to measure the reflectance function is to use a photo-goniometer. This simple instrument can position a small flat sample in any orientation. By recording the radiance for a given surface orientation (p, q) one can obtain the value of reflectance for one point on the reflectance map. Repeating the process for all orientations (p, q) determines the entire reflectance map. These measurements are extremely time-consuming when made manually and difficult to make with any degree of precision. An effort has been made to instrument the goniometer so that reflectance measurements can be gathered automatically by a computer [Ammar, 1978].

To avoid the need to physically move the sample into all possible orientations, one can instead use a test object which presents all possible orientations. The simplest object to use is a sphere. One can then obtain an image with fixed source and viewer (*i.e.* fixed phase angle g). The local surface orientation at a point in the image can be determined by simple trigonometry and paired with the recorded intensity values at that point in the image. Obtaining all orientation-intensity pairs is equivalent to specifying the reflectance map for

the given source and view vectors.

Regardless of how the reflectance map is obtained it is important to remember that it gives scene radiance as a function of local surface orientation (p, q) in a viewer-centered coordinate system.

3. CURRENT METHODS

Before giving the details of the numerical algorithm, we outline several related approaches. They provide a foundation for constructing the numerical scheme as well as a means for comparison.

3.1 The Analytic Approach

Perhaps the most important work in shape-from-shading is due to Horn. His approach attempts to recover the surface shape from a single image by explicitly solving the differential equations of image illumination [Horn, 1970] and [Horn, 1975].

3.1.1 The Set-up

First define the following quantities:

Let the object irradiance at the surface point (x, y, z) be denoted by $a(x, y, z)$. For physical systems, $a(x, y, z)$ is constant or obeys some inverse-square law with respect to distance from the source.

Let t be the ratio of image irradiance to scene radiance. This is a constant which depends on the imaging system.

Let $A(x, y, z) = t \cdot a(x, y, z)$.

Let $\mathbf{r} = (x, y, z)$ be a visible point on the object and $\mathbf{r'} = (x', y', f)$ be the corresponding point in the image, according to the geometry of projection (not necessarily orthographic).

Let $b(x', y')$ be the image irradiance measured at the image point (x', y').

Since scene radiance is proportional to image irradiance, we have

$$A(\mathbf{r}) \; \phi(I, E, G) = b(\mathbf{r'}) \tag{3.1}$$

To show that this equation is a first-order partial differential equation we note that it contains terms involving only x, y, z and the first partial derivatives p and q. This will become apparent in the following:

When finding a solution it is assumed that the object irradiance a(r) and the reflectance function ϕ(I, E, G) are known and the image irradiance b(r') is obtained from the image. From the perspective projection outlined in Section 2.2.1, we have

$$r' = \frac{f}{z} \, r$$

So r' is a function of x, y, and z only. Previously we defined n = (p, q, -1) as the normal to the surface at the point r = (x, y, z). Let the light source be at r_s = (x_s, y_s, z_s). The incident ray can then be defined as r_i = r - r_s and the emergent ray as r_e = r because the viewer is at the origin. Then we have

$$I = \hat{n} \cdot \hat{r}_i \qquad E = \hat{n} \cdot \hat{r}_e \qquad G = \hat{r}_i \cdot \hat{r}_e$$

where the circumflex denotes the unit vector. Inspection reveals that I, E, and G involve only the variables x, y, z, p, q. The general illumination equation can thus be rewritten as

$$A(r) \, \phi(I, E, G) - b(r') = F(x, y, z, p, q) = 0 \tag{3.2}$$

namely, a first-order, non-linear, partial differential equation.

3.1.2 Solution

As Horn points out [Horn, 1975, p.123], the usual method of dealing with a first-order, non-linear, partial differential equation is to solve an equivalent set of five ordinary differential equations:

$$\dot{x} = F_p \qquad \dot{y} = F_q \qquad \dot{z} = p\,F_p + q\,F_q$$

$$\dot{p} = -F_x - p\,F_z \qquad\qquad \dot{q} = -F_y - q\,F_z \tag{3.3}$$

Here the dot denotes differentiation with respect to a parameter s and the subscripts denote partial differentiation. These equations can be solved by the method of characteristic strips [Garabedian, 1967]. The set of equations (3.3) can be understood more fully by making use of the surface Hessian matrix **H** [Woodham, 1978a]. Letting z=f(x, y) we define

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f(x, y)}{\partial x^2} & \dfrac{\partial^2 f(x, y)}{\partial x\, \partial y} \\[2ex] \dfrac{\partial^2 f(x, y)}{\partial x\, \partial y} & \dfrac{\partial^2 f(x, y)}{\partial y^2} \end{bmatrix} \tag{3.4}$$

as the surface Hessian matrix in two dimensions.

The three assumptions of distant viewer, orthographic projection and constant object irradiance allow the basic equation of image formation (3.1) to be simplified and rewritten as

$$R(p, q) = I(x, y) \tag{3.5}$$

Partially differentiating Equation (3.5) with respect to x and y results in two new equations:

$$I_x = p_x\,R_p + q_x\,R_q$$

$$I_y = p_y\,R_p + q_y\,R_q \tag{3.6}$$

For a smooth surface, $p_y = q_x$, so there are two equations relating the three unknowns p_x, q_y, and $p_y=q_x$ where

$$P_x = \frac{\partial^2 z}{\partial x^2}; \quad p_y = q_x = \frac{\partial^2 z}{\partial x \partial y}; \quad q_y = \frac{\partial^2 z}{\partial y^2}$$

Equations (3.6) can be rewritten as the matrix equation

$$\begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} p_x & q_x \\ p_y & q_y \end{bmatrix} \begin{bmatrix} R_p \\ R_q \end{bmatrix} \tag{3.7}$$

Noting that the matrix in the equation above is actually the Hessian matrix, Equation (3.7) can be rewritten

$$[I_x \ I_y]^T = H [R_p \ R_q]^T \tag{3.8}$$

We can relate small movements in the image to the corresponding small movements in gradient space by considering infinitessimal displacements:

$$dp = p_x \, dx + p_y \, dy$$
$$dq = q_x \, dx + q_y \, dy \tag{3.9}$$

Again, two equations can be rewritten as the single matrix equation

$$[dp \ dq]^T = H [dx \ dy]^T \tag{3.10}$$

because the Hessian matrix is symmetric.

Note that while Equation (3.8) is exact at any image point and its corresponding gradient point, Equation (3.10) is only approximate if the steps are of finite size. The Hessian matrix H varies with x and y. It is assumed that for a smooth surface $z=f(x, y)$, third and higher order derivatives are small and can be ignored locally. Then, it is assumed . that [dx dy] can be chosen small enough so that H can be considered constant over the interval from (x, y) to (x+dx, y+dy).

Horn proceeds to find the solution to Equations (3.3) when the projection is

orthographic as follows. Suppose image point (x, y) is known to correspond to a point (p, q) in gradient space. Then, the change in $z=f(x, y)$ corresponding to a small movement $[dx \; dy]$ in the image is given by the following approximation valid for small movements:

$$dz \;\; = \;\; p \, dx \; + \; q \, dy \tag{3.11}$$

The new gradient point $(p + dp, q + dq)$ corresponding to the image point $(x + dx, y + dy)$ is obtained by updating the current gradient (p, q) according to Equation (3.10) If $\mathbf{H}$ were known, the solution could be obtained by iterating this process. This would trace out a path in the image for which the corresponding gradients could be determined. The shape could then be recovered by integrating the gradients along these paths. Unfortunately, there is not enough information to determine the Hessian matrix $\mathbf{H}$.

However, $\mathbf{H}$ is constrained to satisfy Equation (3.8) The solution is continued by exploiting the fact that matrix multiplication is a linear operation. If $[dx \; dy]$ is chosen to be in the direction of $[R_p \; R_q]$ then linearity is sufficient to guarantee that $[dp \; dq]$ will be in the direction of $[I_x \; I_y]$. In mathematical terms, if $[dx \; dy] = [R_p \; R_q] \, ds$, then $[dp \; dq] = [I_x \; I_y] \, ds$ using Equations (3.8 and 3.10). Thus by starting at some known point and iterating these two operations, a path in the image is traced out along which the corresponding gradients, and hence the corresponding relief profile of the object's surface can be determined. The catch is that the direction for $[dx \; dy]$ cannot be chosen arbitrarily. It must be in the direction $[R_p \; R_q]$. The curves traced out on the surface in this fashion are called *characteristics* and their projection in the image plane are called *base characteristics*.

This result has a curious geometric interpretation. Choosing $[dx \; dy]$ to be in the direction of $[R_p \; R_q]$ means that a base characteristic is traced out that is always

perpendicular to the reflectance map contour for the current (p, q). Similarly, the fact that the resulting [dp dq] is in the direction of $[I_x \ I_y]$ means that the corresponding path traced out in gradient space is always perpendicular to the contour of constant image intensity for the current image point (x, y). Unfortunately, the path traced out by the base characteristic depends on the surface being imaged and in general cannot be predetermined or chosen arbitrarily. This is undesirable because characteristics spreading out from an initial curve tend to separate and leave large portions of the surface unexplored. Similarly, they may converge on each other. One obtains only a very uneven sampling of the surface of the object.

3.1.3 Initial Conditions

There are two types of initial conditions necessary and they should not be confused. The first is an initial curve of surface gradients. This value is needed to tie together the solutions obtained along each characteristic. How the initial curve is determined is arbitrary, but it must be known for this procedure to apply. Instead of specifying an initial curve, we can perform a second shape-from-shading calculation using a second image taken with a different source position. Then we can combine the two solutions to determine both components of the local surface normal at most points in the image.

An initial curve of gradients along which the base characteristics can be tied together must be known for any reflectance map and any image. Some reflectance maps provide assistance here. For example, the lambertian reflectance map has a global maximum of one located in the direction of the source (p_s, q_s). Thus the local surface normal is

determined uniquely at the brightest image point (provided there is some surface element oriented in the direction of the source). For specular surfaces, the maximum image intensity corresponds to a surface element with an orientation that is half the inclination of the direction to the source. For the reflectance map associated with the scanning electron microscope, the *minimum* intensity value (the darkest point) corresponds to a point with gradient (0, 0). Whenever such a singular point is available, it can be used to start the solution.

The initial curve of surface normals is the first type of initial conditions needed. The second is a result of our representation of surface shape. We have recovered the surface gradient at all surface points but their distance relative to the viewer is as yet undetermined. Recalling that the surface is expressed as $z=f(x, y)$, we find that the values of z can be recovered by integrating the local surface gradients. The constant of integration needed is the actual depth of one particular surface point since all surface points are specified relative to each other. Again this value must be known from other sources. In many instances however, the actual position of every surface point is not needed -- only the orientation (given by the surface gradient) is required.

3.2 Binocular Stereo

Binocular stereo is an entirely different approach to computing shape from images [Marr & Poggio, 1976; Marr & Poggio, 1977]. This method requires two images of the same scene obtained from slightly different viewpoints. By identifying corresponding surface points in each image, one can determine the disparity (the apparent difference in position of

a surface point from one image to the next) of each pair of points. Surface shape can be recovered by triangulation using simple trigonometry. Binocular stereo thus has no need for knowledge of the reflectance map and works well for discontinuous and non-isotropic surfaces. These two points make it applicable in cases where the reflectance map methods are not useful.

There are several problems with this method: Disparity values are available only at surface points which can be precisely identified in both images. This means that one can determine the "shape" only at selected surface points. The best method of interpolating the surface among the known points is an open question. Of course, another drawback to binocular stereo is that it requires acquisition and analysis of two images.

3.3 Photometric Stereo

The technique of determining local surface orientation from several images with the same view angle but different distributions of light sources has been termed *photometric stereo* [Woodham, 1978b; Horn, Woodham and Silver, 1978]. Briefly, the photometric stereo technique is as follows.

Suppose an image $I(x, y)$ has been obtained under a given imaging geometry and assume that image intensity has been normalized with respect to the reflectance map so that

$$I_1(x, y) = R_1(p, q) \tag{3.12}$$

Choose a particular image point (x_0, y_0) with corresponding image intensity $I_1(x_0, y_0) = \alpha_1$. Then equation (3.12) restricts the range of possible points (p_0, q_0) in gradient space that could possibly correspond to image point (x_0, y_0) to lie on the contour $R_1(p, q) = \alpha_1$.

Now imagine a second image $I_2(x, y)$ obtained under the same object-viewer geometry but with a different light source distribution so that image point (x_o, y_o) in the second image corresponds to the same object point as (x_o, y_o) in the first image. Assume that

$$I_2(x, y) = R_2(p, q) \tag{3.13}$$

Again the corresponding image intensity $I_2(x_o, y_o) = \alpha_2$ specifies a contour in gradient space $R_2(p, q) = \alpha_2$ upon which the actual gradient (p_o, q_o) of surface point (x_o, y_o) must lie. Taking both constraints together, the gradient (p_o, q_o) must lie at the intersection of these two contours in gradient space. Typically, the intersection is a finite set of one, two or more points. To resolve the possible ambiguity one may use a third image obtained with a third source position. The actual gradient (p_o, q_o) must the be at the intersection of all three contours as in Figure 4. Repeating this process for every image point yields the local surface orientation for every corresponding surface point. In practice, it may be necessary to use four sources to guarantee that every gradient point lies in the non-shadowed region of at least three sources. Otherwise, ambiguities could not be resolved.

Photometric stereo has the distinct advantage of being a fast, local computation. One can imagine implementing the technique using table lookup. Under this scheme, every possible triple (or n-tuple) of image intensities $(\alpha_1, \alpha_2, \alpha_3)$ would have associated with it the unique gradient point (p, q) associated with that triple. Of course, many intensity triples would be impossible in practice and would correspond to blanks in the table.

Some limitations on photometric stereo exist. The vectors from the object point to the source must be non-coplanar. Otherwise, the contours of constant intensity

corresponding to each of the three or more image points may intersect in more than one gradient space point, leaving the ambiguity unresolved. This is no problem in an industrial application (for example) where the experimenter can position the light sources at will. However, one cannot control the position of the light source for the purpose of obtaining satellite photos, because the sun appears to travel nearly in a plane with respect to a point on the surface of a planet.
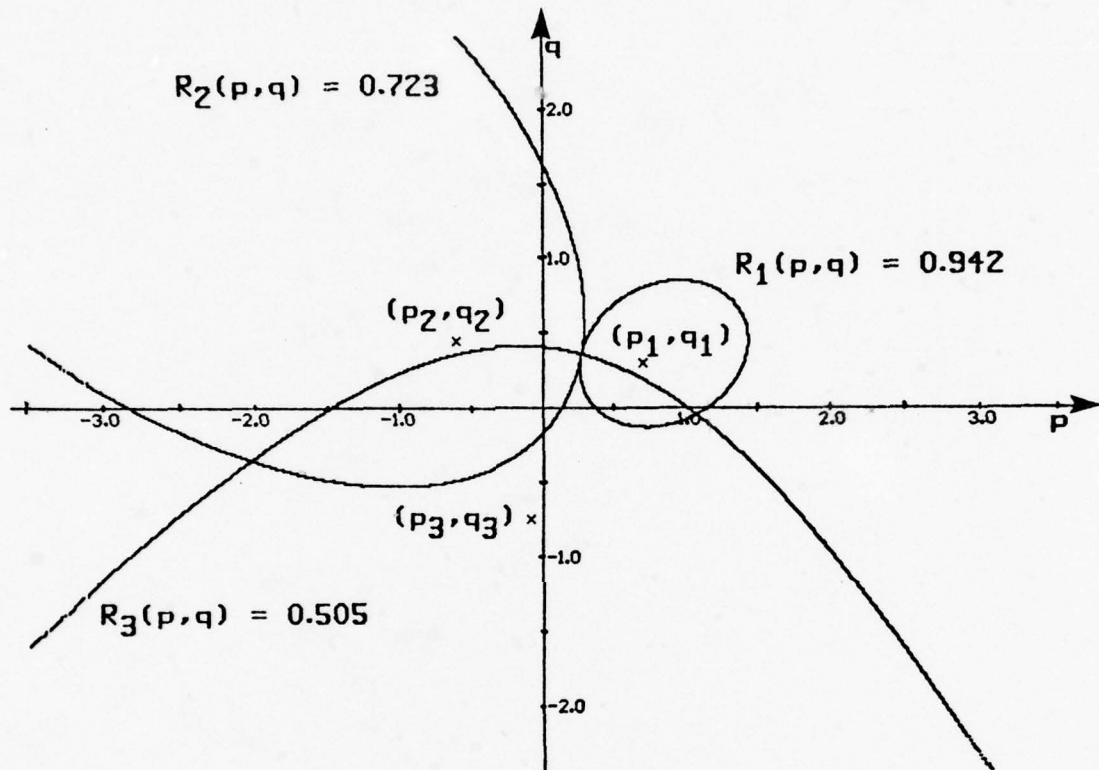
Figure 4 Photometric Stereo

Three reflectance map contours are superimposed. Each contour corresponds to the intensity, value at (u, v) obtained from three separate images taken under the same imaging geometry but with different light source positions. The local surface orientation of (u, v) is at the intersection of all three contours. Here $I_1(u, v) = 0.942$; $I_2(u, v) = 0.723$ and $I_3(u, v) = 0.505$. The surface is assumed to be lambertian and the light sources at (0.7, 0.3), (-0.610, 0.456) and (-0.90, -0.756) respectively. Reprinted from [Woodham, 1978b, p. 18].

4. THE RELAXATION METHOD

We are now ready to give the details of a relatively simple algorithm for performing the shape-from-shading calculation. As noted in the last chapter, all shape-from-shading algorithms possess a number of shortcomings or critical restrictions. The method we pursue here slightly reduces the number of restrictions necessary. It is similar to the cooperative algorithm of Barrow and Tenenbaum [1979], but is not restricted to quadratic surfaces. In the course of the development, the following four requirements are satisfied:

The shape-from-shading algorithm is to determine surface shape from a single image.

The image may be of any *smooth* surface with constant photometric properties.

The algorithm must be applicable for any well-behaved reflectance map.

The surface orientation must be determined at all points in the image.

4.1 Constraints

Many complex systems can be analyzed by isolating their inherent constraints. Image analysis is no exception. One constraint on the image irradiance is provided by the reflectance properties of the surfaces. Restricting attention to smooth surfaces provides another constraint. Together, these two constraints provide enough information to recover surface shape from a single image.

4.1.1 Image Intensity

Recall Horn's basic equation of image formation:

$$A(r) \; \phi(i, e, g) = b(r')$$ (4.1)

Let us restrict attention to those situations in which

(1) the light source is distant;

(2) the projection is orthographic mapping object point (x, y, z) into image point (x, y); and

(3) each surface point receives the same incident illumination (irradiance).

The last restriction implies that $A(r)$ is constant. Orthographic projection allows one to write

$$b(r') = I(x, y)$$ (4.2)

where $I(x, y)$ is the intensity value recorded in an image and is not to be confused with I, the cosine of the incident angle. Finally, $\phi(I, E, G)$ can be rewritten as $R(p, q)$ since restriction (1) implies that G is constant. If the reflectance map is normalized with respect to the intensities recorded in the image, then the image forming equation becomes

$$R(p, q) = I(x, y)$$ (4.3)

In other words, scene radiance must equal image irradiance. As in photometric stereo, an intensity value α_1 recorded at image point (x, y) restricts the possible range of values of the local surface normal at the object point corresponding to (x, y) to lie on the one-parameter contour of the reflectance map which satisfies $R(p, q) = \alpha_1$ in gradient space.

4.1.2 Surface Smoothness

The constraint supplied by the intensities recorded in an image enabled us to restrict the range of possible gradients at a given point to within one degree of freedom. We have apparently used all the information contained in the image intensities, so from where does the restriction on the other degree of freedom come?

We have chosen to represent the surface shape by the value of the gradient at each image point. The gradients are partial derivatives of the surface $z=f(x, y)$ so they must integrate to a unique surface if the surface they represent is real. Thus an arbitrary assignment of gradients at image points may not necessarily represent a realizable smooth surface even when the gradients lie on the appropriate contours of the reflectance map. The last degree of freedom can be eliminated by presupposing that the surface is smooth.

<u>Theorem</u> For any smooth surface $z=f(x, y)$,

$$\frac{\partial^2 z}{\partial x\, \partial y} = \frac{\partial^2 z}{\partial y\, \partial x}$$

Or equivalently

$$\frac{\partial p}{\partial y} = \frac{\partial q}{\partial x} \tag{4.4}$$

Equation (4.4) provides the additional dimension of constraint. Another way of expressing this constraint of surface smoothness is

$$\oint n \cdot ds = 0 \tag{4.5}$$

where $n = (p, q)$ the gradient at a given point and $ds = (dx, dy)$ is an infinitessimal line element on the surface. Expanding the dot product yields

$$\oint p\, dx + q\, dy = 0 \tag{4.6}$$

which is finally in a form that can be utilized.

4.2 Implementation of the Constraints

Two constraints have been delineated. The first, $I(x, y) = R(p, q)$ is strictly local. For any given image point, the range of possible gradients is restricted to a contour of gradient space. On the other hand, the smoothness constraint is not local. The feasibility of a gradient (p, q) at point (x, y) is dependent upon the gradients of the neighbors of (x, y). What we seek is an iterated local computation which enforces these constraints at all image points. If all works as planned, the computation will be "pseudo-local" and ultimately converge to a global solution.

To study this approach, we define an error function ϵ which measures the "distance" that a given assignment of surface orientations is from the solution. This error function will be separated into two parts, one corresponding to each constraint. Letting ϵ_s be a measure of the departure from surface smoothness and ϵ_r be a measure of the departure from the basic equation of image formation, the following equation is proposed.

$$\epsilon = \epsilon_s + \rho \, \epsilon_r \tag{4.7}$$

Here, ρ is a scale factor to bring the arbitrary units of the error functions ϵ_s and ϵ_r in line. It will now be shown how these error functions are determined.

4.2.1 Image Intensity

The factor ϵ_r is to be a measure of the departure of the present estimate of the solution from the image according to the image forming equation. For the current

assignment of gradient (p, q) at an image point, a value R(p, q) can be calculated using the reflectance map. This value represents the intensity that would be recorded in an image *if* the assignment of (p, q) were correct at the corresponding object point. The "distance" of the actual image intensity from the predicted intensity is the quantity

$$I(x, y) - R(p, q)$$

Therefore, the equation

$$\epsilon_r = [I(x, y) - R(p, q)]^2 \tag{4.8}$$

which restricts the error measure to non-negative values, appears to be a reasonable choice. This equation has two desirable properties. First, when a proposed gradient (p, q) lies on the particular reflectance map contour as determined by the image, the error $\epsilon_r = 0$. The second property is that any other value of (p, q) results in a positive value of ϵ_r and the further the value of R(p, q) from the image intensity I(x, y), the more positive the value of ϵ_r. Therefore, the constraint imposed by the imaging process can be enforced by minimizing Equation (4.8) at every image point.

4.2.2 Surface Smoothness

The factor ϵ_s is to be a measure of departure from surface *smoothness*. Note that we will not be seeking the smoothest surface that could have given rise to a particular image but any surface that possesses second partial derivatives everywhere subject to the constraints of the imaging equation. In this section, two representations for ϵ_s are derived. The first represents a simple-minded approach whereas the second embodies a more complicated and more desirable technique. Both take advantage of several heuristics.

4.2.2.1 The Simple Way

The simple way implements Equation (4.4) directly. A first-order approximation for $\partial p / \partial y$ at point (u, v) can be expressed as

$$\partial p / \partial y = p_{u(v+1)} - p_{uv} \qquad (4.9)$$

Here, the abbreviation p_{uv} is taken to mean the value of p at the surface point corresponding to image point (u, v). Similarly,

$$\partial q / \partial x = q_{(u+1)v} - q_{uv} \qquad (4.10)$$

Substituting (4.9) and (4.10) into Equation (4.4) yields

$$p_{u(v+1)} - p_{uv} = q_{(u+1)v} - q_{uv}$$

Therefore, if $z=f(x, y)$ is a smooth surface, it must be the case that

$$p_{u(v+1)} - p_{uv} - q_{(u+1)v} + q_{uv} = 0 \qquad (4.11)$$

Thus, a good measure of departure from surface smoothness at point (u, v) is

$$\epsilon_s = [p_{u(v+1)} - p_{uv} - q_{(u+1)v} + q_{uv}]^2 \qquad (4.12)$$

Note however, that this estimate of departure from smoothness takes into consideration only points that are above or to the right of point (u, v). For symmetry, estimates can be made for all four quadrants around point (u, v) as illustrated in Figure 5. The equations are abbreviated to show their dependence on p_{uv} and q_{uv} in order to facilitate the partial differentiations which will eventually be performed.

We are now in a position to construct an expression for ϵ_s. All four error expressions ϵ_A, ϵ_B, ϵ_C and ϵ_D are to be minimized, so summing them seems to be an appropriate thing to do.

(u, v+1)

(u-1, v)   (u, v)
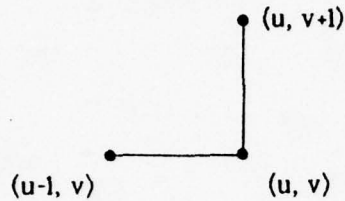
$q_{uv} - q_{(u-1)v} = p_{u(v+1)} - p_{uv}$
$q_{uv} - q_{(u-1)v} - p_{u(v+1)} + p_{uv} = 0$
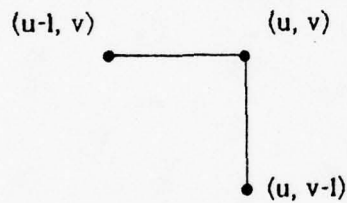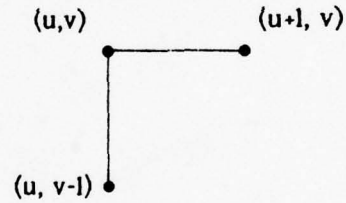Let $D = q_{(u-1)v} - p_{u(v+1)}$
$D + p_{uv} + q_{uv} = 0$
$\epsilon_D = (D + p_{uv} + q_{uv})^2$

(u, v+1)

(u,v)   (u+1, v)
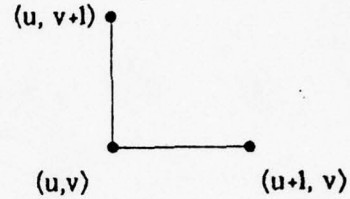
$q_{(u+1)v} - q_{uv} = p_{u(v+1)} - p_{uv}$
$q_{(u+1)v} - q_{uv} - p_{u(v+1)} + p_{uv} = 0$
Let $A = q_{(u+1)v} - p_{u(v+1)}$
$A + p_{uv} - q_{uv} = 0$
$\epsilon_A = (A + p_{uv} - q_{uv})^2$

(u-1, v)   (u, v)

(u, v-1)

$q_{uv} - q_{(u-1)v} = p_{uv} - p_{u(v-1)}$
$q_{uv} - q_{(u-1)v} - p_{uv} + p_{u(v-1)} = 0$
Let $C = p_{u(v-1)} - q_{(u-1)v}$
$C - p_{uv} + q_{uv} = 0$
$\epsilon_C = (C - p_{uv} + q_{uv})^2$

(u,v)   (u+1, v)

(u, v-1)

$q_{(u+1)v} - q_{uv} = p_{uv} - p_{u(v-1)}$
$q_{(u+1)v} - q_{uv} - p_{uv} + p_{u(v-1)} = 0$
Let $B = q_{(u+1)v} - p_{u(v-1)}$
$B - p_{uv} - q_{uv} = 0$
$\epsilon_B = (B - p_{uv} - q_{uv})^2$

Figure 5 The Simple Measure of Surface Smoothness.
This figure shows the calculation of the error functions ϵ_A, ϵ_B, ϵ_C and ϵ_D in each of the four quadrants. Together they enforce the assumption of local surface smoothness at image point (u, v).

$$\epsilon_s = \epsilon_A + \epsilon_B + \epsilon_C + \epsilon_D$$

$$= (A+p-q)^2 + (B-p-q)^2 + (C-p+q)^2 + (D+p+q)^2 \qquad (4.13)$$

where the subscripts have been eliminated from p_{uv} and q_{uv} for convenience.

It will now be shown that this formalization of the smoothness constraint is perhaps not appropriate. At what gradient point is ϵ_s minimized? Partial differentiation followed by evaluation at zero will answer this question.

$$\partial \epsilon_s / \partial p_{uv} = 2(A+p-q) - 2(B-p-q) - 2(C-p+q) + 2(D+p+q) = 0$$

$$\partial \epsilon_s / \partial q_{uv} = -2(A+p-q) - 2(B-p-q) + 2(C-p+q) + 2(D+p+q) = 0$$

So

$$2A - 2B - 2C + 2D + 8p = 0$$

$$-2A - 2B + 2C + 2D + 8q = 0$$

Therefore

$$p = \tfrac{1}{4}(-A + B + C - D)$$

$$q = \tfrac{1}{4}(A + B - C - D) \qquad (4.14)$$

Restoring the abbreviations and subscripts gives

$$p_{uv} = \tfrac{1}{2}(p_{u(v+1)} + p_{u(v-1)})$$

$$q_{uv} = \tfrac{1}{2}(q_{(u+1)v} + q_{(u-1)v}) \qquad (4.15)$$

Therefore, p_{uv} is seen to depend only on p values directly above and below the image point (u, v). Similarly, q_{uv} depends only on values of q directly to the right and left. This decoupling of p and q implies that adjacent columns of p values, like adjacent rows of q

values, are independent. Somehow, the notion of surface smoothness as implied by Equation (4.4) seems not to have been captured. It ought to be the case that a gradient (p, q) at point (u, v) should be affected by the gradients of all its immediate neighbors. This formalism, which updates by averaging within a single row or column, apparently permits such undesirable conditions. Experimentation has shown that implementation of the numerical shape-from-shading algorithm using Equations (4.15) does not produce convergence in general.

We must do better.

4.2.2.2 A More Complicated Technique

The development in this section parallels the development of the last section exactly. The insight gained there will help keep things straight as they start to get lengthy here. The theory is not difficult -- it has not changed since the last section. The equations grow long but it is important to understand what they imply. Abbreviations will again be used where applicable.

This technique utilizes the smoothness constraint as expressed in Equation (4.6)

$$\oint \mathbf{n} \cdot \mathbf{ds} = \oint p\,dx + q\,dy = 0 \tag{4.16}$$

The line integral is to be evaluated around a "loop" as shown in Figure 6. Here the *loop* takes the form of a square with sides of length equal to one grid point in the image. Evaluating Equation (4.16) along each side of the square gives

$$\oint \mathbf{n} \cdot \mathbf{ds} = \int_u^{u+1} p_{xv}\,dx + \int_v^{v+1} q_{(u+1)y}\,dy + \int_{u+1}^u p_{x(v+1)}\,dx + \int_{v+1}^v q_{uy}\,dy$$

$$\int_{u+1}^{u} p_{x(v+1)}\, dx$$
$$= \tfrac{1}{2}\,[p_{(u+1)(v+1)} + p_{u(v+1)}]$$

(u, v+1)   (u+1, v+1)

$$\int_{v}^{v+1} q_{(u+1)y}\, dy$$
$$= \tfrac{1}{2}\,[q_{u(v+1)} + q_{uv}]$$

$$\int_{v+1}^{v} q_{uy}\, dy$$
$$= \tfrac{1}{2}\,[q_{(u+1)v} + p_{(u+1)(v+1)}]$$

(u, v)   (u+1, v)

$$\int_{u}^{u+1} p_{xv}\, dx$$
$$= \tfrac{1}{2}\,[p_{uv} + p_{(u+1)v}]$$

$$\oint \mathbf{n}\cdot d\mathbf{s} = \int_{u}^{u+1} p_{xv}\, dx \;+\; \int_{v}^{v+1} q_{(u+1)y}\, dy \;+\; \int_{u+1}^{u} p_{x(v+1)}\, dx \;+\; \int_{v+1}^{v} q_{uy}\, dy$$

$$= \tfrac{1}{2}\,[p_{uv} + p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)} - p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)} - q_{uv}]$$

$$\epsilon_s = [p_{uv} + p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)} - p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)} - q_{uv}]^2$$

Figure 6 Enforcing Surface Smoothness Around a Loop

This figure shows the derivation of the approximation used to estimate the local departure from surface smoothness at image point (u, v). The equation is exact for quadratic surfaces and will be taken as a good approximation for non-quadratic surfaces.

If the surface is assumed to be piecewise quadratic, then the following integral is exact.

$$\int_u^{u+1} p_{xv}\, dx = \tfrac{1}{2}\left(p_{uv} + p_{(u+1)v}\right) \tag{4.17}$$

For non-quadratic surfaces, it will be taken to be a close approximation. Being careful with minus signs we obtain the approximation

$$\oint n \cdot ds = \tfrac{1}{2}\left[p_{uv} + p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)} - p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)} - q_{uv}\right] \tag{4.18}$$

Thus the departure from surface smoothness can be expressed as

$$\epsilon_s = \left[p_{uv} + p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)} - p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)} - q_{uv}\right]^2$$

As in the last section, this expression involves only p and q values that lie above and to the right of point (u, v). For symmetry, all four quadrants must again be considered. Figure 7 shows the resulting forms. The letters A, B, C and D have again been used as abbreviations but they are different from the A, B, C and D of the last section. The remainder of this thesis deals only with the A, B, C and D expressions from this section as shown in Figure 7.

Finally, we arrive at an expression for e_s as in the last section.

$$\epsilon_s = \epsilon_A + \epsilon_B + \epsilon_C + \epsilon_D$$
$$= (A+p-q)^2 + (B-p-q)^2 + (C-p+q)^2 + (D+p+q)^2 \tag{4.19}$$

Again, the gradient (p, q) which minimizes the departure from surface smoothness ϵ_s can be found by differentiating and evaluating at zero. Noting that this equation is identical to Equation (4.13) except for the abbreviations A, B, C and D, allows one to write the answer immediately from Equation (4.14).

$$p = \tfrac{1}{4}(-A + B + C - D)$$
$$q = \tfrac{1}{4}(A + B - C - D) \tag{4.20}$$

Top-left diagram labels: (u-1, v+1), (u, v+1), (u-1, v), (u, v)
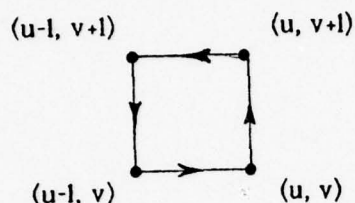
$$\frac{1}{2}[p_{(u-1)v} + p_{uv} + q_{uv} + q_{u(v+1)}$$
$$- p_{u(v+1)} - p_{(u-1)(v+1)} - q_{(u-1)(v+1)} - q_{(u-1)v}] = 0$$
$$\text{Let } D = p_{(u-1)v} + q_{u(v+1)} - p_{u(v+1)}$$
$$- p_{(u-1)(v+1)} - q_{(u-1)(v+1)} - q_{(u-1)v}$$
$$\text{Then } D + p_{uv} + q_{uv} = 0$$
$$\epsilon_D = (D + p_{uv} + q_{uv})^2$$

Top-right diagram labels: (u, v+1), (u+1, v+1), (u, v), (u+1, v)
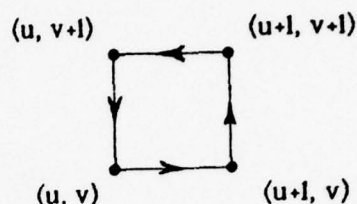
$$\frac{1}{2}[p_{uv} + p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)}$$
$$- p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)} - q_{uv}] = 0$$
$$\text{Let } A = p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)}$$
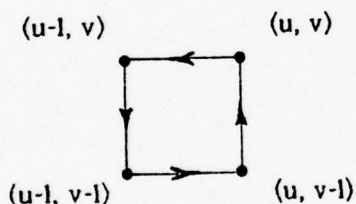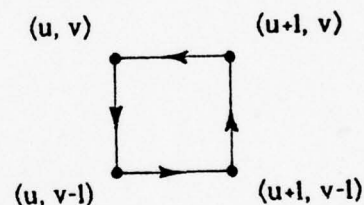$$- p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)}$$
$$\text{Then } A + p_{uv} - q_{uv} = 0$$
$$\epsilon_A = (A + p_{uv} - q_{uv})^2$$

Bottom-left diagram labels: (u-1, v), (u, v), (u-1, v-1), (u, v-1)

$$\frac{1}{2}[p_{(u-1)(v-1)} + p_{u(v-1)} + q_{u(v-1)} + q_{uv}$$
$$- p_{uv} - p_{(u-1)v} - q_{(u-1)v} - q_{(u-1)(v-1)}] = 0$$
$$\text{Let } C = p_{(u-1)(v-1)} + p_{u(v-1)} + q_{u(v-1)}$$
$$- p_{(u-1)v} - q_{(u-1)v} - q_{(u-1)(v-1)}$$
$$\text{Then } C - p_{uv} + q_{uv} = 0$$
$$\epsilon_C = (C - p_{uv} + q_{uv})^2$$

Bottom-right diagram labels: (u, v), (u+1, v), (u, v-1), (u+1, v-1)

$$\frac{1}{2}[p_{u(v-1)} + p_{(u+1)(v-1)} + q_{(u+1)(v-1)} + q_{(u+1)v}$$
$$- p_{(u+1)v} - p_{uv} - q_{uv} - q_{u(v-1)}] = 0$$
$$\text{Let } B = p_{u(v-1)} + p_{(u+1)(v-1)} + q_{(u+1)(v-1)}$$
$$+ q_{(u+1)v} - p_{(u+1)v} - q_{u(v-1)}$$
$$\text{Then } B - p_{uv} - q_{uv} = 0$$
$$\epsilon_B = (B - p_{uv} - q_{uv})^2$$

Figure 7 The Four Loops Measure of Surface Smoothness
This figure shows the calculation of the estimate of departure from local surface smoothness at image point (u, v). The approximation is made from all four quadrants for symmetry.

Equation (4.20) was found to be inadequate with the old definitions of A, B, C and D. Is it now satisfactory?

By writing Equations (4.20) out in full and rearranging terms, Equations (4.21) are found.

$$
\begin{aligned}
p_{uv} = \frac{1}{4} \{ & [p_{(u-1)(v-1)} + p_{(u+1)(v-1)} + p_{(u+1)(v+1)} + p_{(u-1)(v+1)}] \\
& + 2[p_{u(v-1)} + p_{u(v+1)} - p_{(u-1)v} - p_{(u+1)v}] \\
& + [q_{(u-1)(v+1)} + q_{(u+1)(v-1)} - q_{(u-1)(v-1)} - q_{(u+1)(v+1)}] \}
\end{aligned}
$$

$$
\begin{aligned}
q_{uv} = \frac{1}{4} \{ & [q_{(u-1)(v-1)} + q_{(u+1)(v-1)} + q_{(u+1)(v+1)} + q_{(u-1)(v+1)}] \\
& - 2[q_{u(v-1)} + q_{u(v+1)} - q_{(u-1)v} - q_{(u+1)v}] \\
& + [p_{(u-1)(v+1)} + p_{(u+1)(v-1)} - p_{(u-1)(v-1)} - p_{(u+1)(v+1)}] \}
\end{aligned}
$$

(4.21)

Fortunately, these equations can be understood by looking at the templates of Figure 8. These templates are first centered over point (u, v), then the template values are multiplied by the corresponding p or q values, and finally, these products are summed. These equations show that p and q depend on all sixteen neighboring values as hoped.

Again we should check if these expressions make sense. Suppose the gradient at image point (u, v) is known to be (p^*, q^*). Now define

$$
r = \frac{\partial^2 z}{\partial x^2} = \frac{\partial p}{\partial x} ; \quad s = \frac{\partial^2 z}{\partial x \partial y} = \frac{\partial p}{\partial y} = \frac{\partial q}{\partial x} ; \quad t = \frac{\partial^2 z}{\partial y^2} = \frac{\partial q}{\partial y}
$$

Assume that z=f(x, y) can be approximated by a piecewise quadratic surface. For quadratic surfaces, the derivatives higher than second order are zero. Therefore, the following values can be written.

p_{uv}^{i+1}

p_{uv}^{i} q_{uv}^{i}

$$p_{uv}^{i+1} = \frac{1}{4} \{ [p_{(u-1)(v-1)}^{i} + p_{(u+1)(v-1)}^{i} + p_{(u+1)(v+1)}^{i} + p_{(u-1)(v+1)}^{i}]$$
$$+ 2[p_{u(v-1)}^{i} + p_{u(v+1)}^{i} - p_{(u-1)v}^{i} - p_{(u+1)v}^{i}]$$
$$+ [q_{(u-1)(v+1)}^{i} + q_{(u+1)(v-1)}^{i} - q_{(u-1)(v-1)}^{i} - q_{(u+1)(v+1)}^{i}] \}$$

q_{uv}^{i+1}

p_{uv}^{i} q_{uv}^{i}

$$q_{uv}^{i+1} = \frac{1}{4} \{ [q_{(u-1)(v-1)}^{i} + q_{(u+1)(v-1)}^{i} + q_{(u+1)(v+1)}^{i} + q_{(u-1)(v+1)}^{i}]$$
$$- 2[q_{u(v-1)}^{i} + q_{u(v+1)}^{i} - q_{(u-1)v}^{i} - q_{(u+1)v}^{i}]$$
$$+ [p_{(u-1)(v+1)}^{i} + p_{(u+1)(v-1)}^{i} - p_{(u-1)(v-1)}^{i} - p_{(u+1)(v+1)}^{i}] \}$$
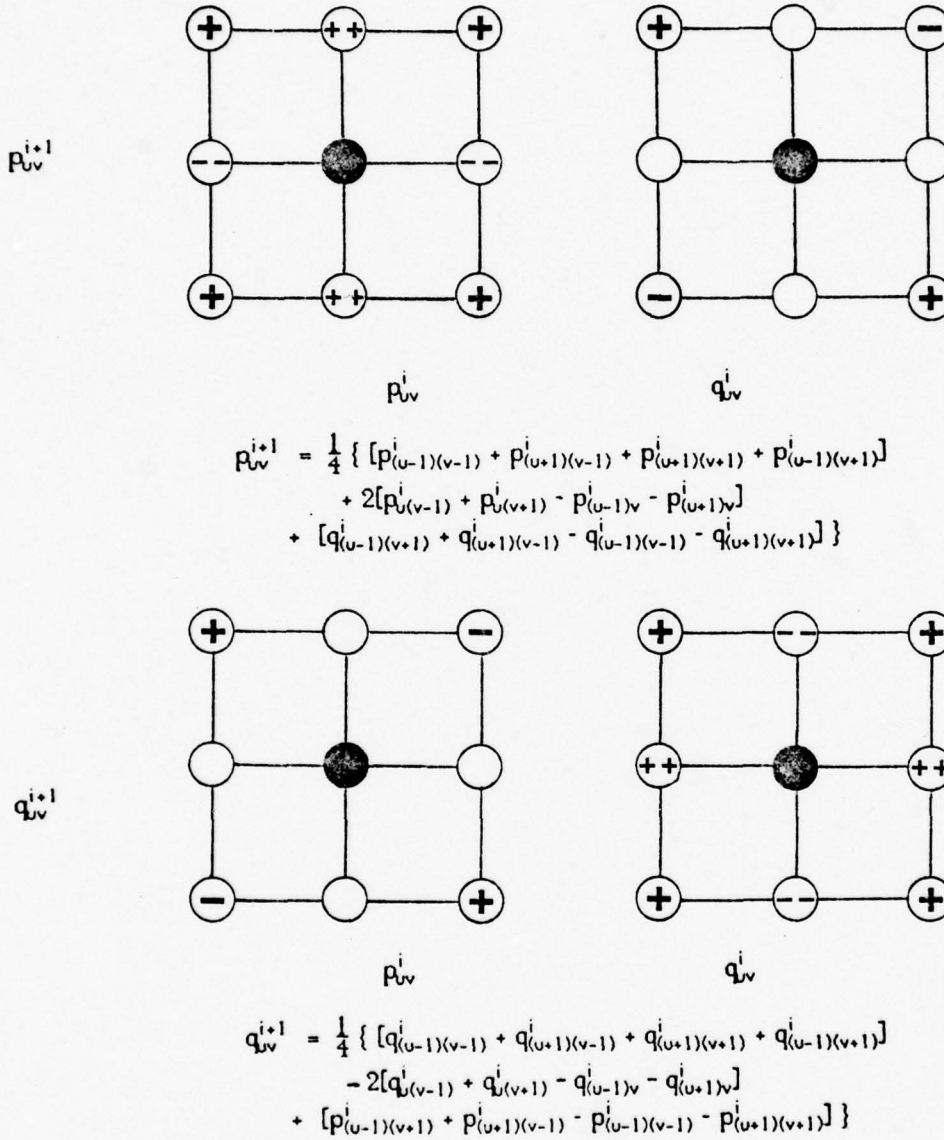
Figure 8 Templates of Local Smoothness Operators

The top pair illustrates the computation for updating p_{uv} and the lower pair illustrates the computation that updates q_{uv}. The left templates are postioned over the current p-values while the right ones act on q-values. New gradient values are computed by summing corresponding p and q values according to the templates. Each sum is divided by four to obtain the optimal gradient at point (u, v) given the gradients of the neighbors.

$$p_{(u-1)(v-1)} = p^* + \Delta(-r-s) \qquad q_{(u-1)(v-1)} = q^* + \Delta(-s-t)$$
$$p_{(u-1)(v+1)} = p^* + \Delta(-r+s) \qquad q_{(u-1)(v+1)} = q^* + \Delta(-s+t)$$
$$p_{(u+1)(v+1)} = p^* + \Delta(+r+s) \qquad q_{(u+1)(v+1)} = q^* + \Delta(+s+t)$$
$$p_{(u+1)(v-1)} = p^* + \Delta(+r-s) \qquad q_{(u+1)(v-1)} = q^* + \Delta(+s-t)$$
$$p_{u(v-1)} = p^* + \Delta(-s) \qquad q_{u(v-1)} = q^* + \Delta(-t)$$
$$p_{u(v+1)} = p^* + \Delta(+s) \qquad q_{u(v+1)} = q^* + \Delta(+t)$$
$$p_{(u-1)v} = p^* + \Delta(-r) \qquad q_{(u-1)v} = q^* + \Delta(-s)$$
$$p_{(u+1)v} = p^* + \Delta(+r) \qquad q_{(u+1)v} = q^* + \Delta(+s)$$

where Δ is the distance between adjacent grid points. Using these expressions we find that

$$\tfrac{1}{4}[p_{(u-1)(v-1)} + p_{(u+1)(v-1)} + p_{(u+1)(v+1)} + p_{(u-1)(v+1)}] = \tfrac{1}{4}[4p^*] = p^*$$
$$\tfrac{1}{2}[p_{u(v-1)} + p_{u(v+1)} - p_{(u-1)v} - p_{(u+1)v}] = 0$$
$$\tfrac{1}{4}[q_{(u-1)(v+1)} + q_{(u+1)(v-1)} - q_{(u-1)(v-1)} - q_{(u+1)(v+1)}] = 0$$

and $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ (4.22)

$$\tfrac{1}{4}[q_{(u-1)(v-1)} + q_{(u+1)(v-1)} + q_{(u+1)(v+1)} + q_{(u-1)(v+1)}] = \tfrac{1}{4}[4q^*] = q^*$$
$$\tfrac{1}{2}[q_{u(v-1)} + q_{u(v+1)} - q_{(u-1)v} - q_{(u+1)v}] = 0$$
$$\tfrac{1}{4}[p_{(u-1)(v+1)} + p_{(u+1)(v-1)} - p_{(u-1)(v-1)} - p_{(u+1)(v+1)}] = 0$$

So for a piecewise quadratic surface, we see that our estimate for p_{uv} is the actual value p^* and the estimate for q_{uv} is q^* as expected. Hence it is shown that these equations do satisfy the smoothness constraint and are exact for surfaces that are piecewise quadratic. In addition, it seems desirable that the gradient at point (u, v) depends on both values of all eight immediate neighbors as it does here. For non-quadratic surfaces, Equations (4.21) are a good approximation for minimizing ϵ_s.

Equation (4.19) has been chosen to be the equation used for specifying a measure of departure from surface smoothness.

4.3 Relaxation

An expression for departure from the true shape of an imaged surface is now in hand.

$$\epsilon = \epsilon_s + \rho \, \epsilon_r$$

$$= (A+p-q)^2 + (B-p-q)^2 + (C-p+q)^2 + (D+p+q)^2 + \rho \, [I(x, y) - R(p, q)]^2 \quad (4.23)$$

It has been postulated that the shape of the imaged surface can be recovered if Equation (4.7) can be minimized at all image points simultaneously. That is, we actually have to find a set of values for p and q which minimizes the error

$$E = \sum_u \sum_v \epsilon_{uv} =$$

$$\sum_u \sum_v (A_{uv}+p_{uv}-q_{uv})^2 + (B_{uv}-p_{uv}-q_{uv})^2 + (C_{uv}-p_{uv}+q_{uv})^2 + (D_{uv}+p_{uv}+q_{uv})^2 + \rho \, [I(u, v)-R(p_{uv}, q_{uv})]^2$$

$$(4.24)$$

An *iterative relaxation* scheme is employed to accomplish this. Loosely, this scheme works at two levels. At the lower level, local operators at each image point attempt to locally enforce smoothness on the proposed solution surface while simultaneously satisfying the general illumination equation. In other words, each local operator attempts to minimize Equation (4.7) at its own image point. At the higher level, the interaction of the local operators due to their overlapping with their immediate neighbors propagates some information to those neighbors. By iterating the entire process, a form of global communication is established. The intention is that this propagation of information allows each local operator to gradually determine a value of its local surface orientation such that all local operators simultaneously reach a universal minimum. Before getting into the

mathematics, let's examine these ideas more closely.

4.3.1 The Local Operators

Imagine the local operator as a little machine attached to a particular image point. Its job is to determine values of p and q which simultaneously minimize ϵ_s and ϵ_r according to Equation (4.7). Choosing any gradient (p, q) that lies on the contour $R(p, q) = I(x, y)$ of the reflectance map minimizes ϵ_r; in fact, it makes ϵ_r equal to zero. ϵ_s is minimized at each point (u, v) by Equations (4.21). The question is how to satisfy both constraints simultaneously.

When the reflectance map is a very simple analytic function of p and q, the local operator's job is easy. By differentiating Equation (4.23) and evaluating at zero, a value of the gradient can be found which minimizes Equation (4.7) and hence satisfies both constraints simultaneously.

In most cases of interest, however, the reflectance function is not a simple analytic function -- it may not be analytic at all -- and we must resort to more sophisticated techniques of minimization. The details of three such methods are described in Section 4.4.

4.3.2 Achieving Global Constraint

We now face up to the issue of finding the global minimum of E, given that the local operators are capable of finding a local minimum of ϵ at each image point. To do this, an iterative relaxation scheme is employed. There is a substantial literature on relaxation methods [Allen, 1954; Wilde, 1966], but all are concerned with determining a single function.

We wish to determine two functions (the components of the gradient at each point) where each value depends on both values of its neighbors and on an additional, strictly local, constraint (the intensity value in the image). The relaxation scheme used is similar in principle to the single-function relaxation scheme, but much more difficult to analyze. Here's how it works.

Consider an arbitrary, initial determination of gradients at each image point. Now every local operator looks at its corresponding intensity value in the image and at the gradients of its neighbors and determines a new gradient that minimizes its own ϵ. At this point, the collection of all new gradients so determined defines a new estimate of shape and the old one is forgotten. Again, each local operator determines a new gradient, different from the last gradient it determined because the gradients of its neighbors have changed. Repeating this process indefinitely, we find that the current assessment of surface shape typically converges to a stable assessment. If the total error E of this assessment is near zero, then it must be a smooth object surface that could have given rise to the image, thus solving the shape-from-shading problem.

The explanation of the success of this scheme is not so simple. The neighboring gradients can be thought of as exerting a "force" to turn the local solution in one direction or another. In the massed effect of all eight neighbors of a given point, erroneous forces tend to cancel out while valid forces are in the same direction and compound.

Viewed from a different perspective, one can consider the error function ϵ at a point as a surface in gradient space. Minimizing ϵ corresponds to sliding down this surface to a new gradient point with a lower value of ϵ. When the neighbor's gradients change, the

error surface undulates somewhat, but hopefully has not raised us to a much higher value of ϵ. Then by continually sliding a lot and rising a little, we gradually work our way down to the actual minimum of ϵ.

A proof of the convergence to a globally smooth surface can be found in Section 5.3. For now, the iteration can be viewed as having one local operator for each image point and all operators update their current gradient estimates simultaneously. The overlap of the local operators allows the local constraint to propagate about in the image, thereby achieving global constraint and convergence to the solution to the shape-from-shading problem.

4.4 The Form of the Local Operators

We now return attention to the issue of how to minimize both error functions simultaneously. A search of the literature reveals a number of methods for finding an extremum of a function of several values. Methods using the gradient of the error function (not our surface gradient) include the classical method of steepest descents [Curry, 1944; Householder, 1953], some variations of steepest descents [Levenberg, 1944; Booth, 1957], the "PARTAN" method [Shah, Buehler and Kempthorne, 1961], a more recent method due to Powell [1962], an improvement based on that method [Fletcher and Powell, 1963], and a method based on parameter variation [Deist and Sefor, 1967]. A summary of minimization methods can be found in [Pun, 1969] or [Szego, 1972].

Three alternatives for solution are presented here. As will be seen, the last is the least restrictive. The first two require that the reflectance map be known analytically. The first method, that of Lagrange Multipliers, further requires that the reflectance function be

expressible in a sufficiently simple form.

The mathematics become more complicated here. Fortunately, insightful geometric interpretations are often available.

4.4.1 Lagrange Multipliers

The method of Lagrange Multipliers allows one to find a minimum of the error function ϵ immediately, without iteration. The price to be paid for this will be seen to be that the reflectance map must be explicitly differentiable and the derivatives must be invertible. The method of Lagrange Multipliers as it applies to the error function at hand is as follows.

4.4.1.1 Mathematical Details

First recast Equation (4.23) into two parts. The problem is then to minimize

$$\epsilon_s = (A+p-q)^2 + (B-p-q)^2 + (C-p+q)^2 + (D+p+q)^2 \tag{4.25}$$

subject to the condition

$$R(p, q) - I(x, y) = 0 \tag{4.26}$$

Note that satisfying the constraint of Equation (4.26) is equivalent to forcing $\epsilon_r=0$. Therefore minimizing (4.25) subject to the condition of Equation (4.26) will be approximately equivalent to minimizing

$$\epsilon = \epsilon_s + \rho \, \epsilon_r \tag{4.7}$$

Now form a new equation

$$F = \epsilon_s + \lambda[R(p, q) - I(x, y)] \tag{4.27}$$

where λ is a constant (the Lagrange multiplier). The necessary conditions for F to have stationary points (extrema) are

$$\frac{\partial F}{\partial p} = \frac{\partial \epsilon_s}{\partial p} + \lambda \frac{\partial}{\partial p}[R(p, q) - I(x, y)] = 0$$

and

$$\frac{\partial F}{\partial q} = \frac{\partial \epsilon_s}{\partial q} + \lambda \frac{\partial}{\partial q}[R(p, q) - I(x, y)] = 0$$

(4.28)

Before proceeding further, note that we can recast ϵ_s as

$$\epsilon_s = 4(p-p_0)^2 + 4(q-q_0)^2 + \epsilon_0 \tag{4.29}$$

where ϵ_0 does not depend on p or q and p_0 and q_0 are the p and q of Equation (4.14).

$$p_0 = \frac{1}{4}(-A + B + C - D)$$

$$q_0 = \frac{1}{4}(A + B - C - D)$$

(4.30)

Now solving Equations (4.28) and including Equation (4.26) yields the following set of three equations in the three unknowns p, q and λ.

$$8(p-p_0) + \lambda R_p = 0$$

$$8(q-q_0) + \lambda R_q = 0 \tag{4.31}$$

$$R(p, q) = I(x, y)$$

Eliminating the Lagrange multiplier λ yields

$$(p-p_0) R_q = (q-q_0)R_p \tag{4.32}$$

$$R(p, q) = I(x, y) \tag{4.33}$$

The solution of these equations for p and q yields the desired gradient point that minimizes ϵ_s while satisfying the constraint $R(p, q) = I(x, y)$. Whether or not they can be explicitly solved for p and q depends on the form of $R(p, q)$ and its partial derivatives R_p and R_q.

4.4.1.2 Geometric Interpretation

Recall that the method of Lagrange Multipliers seeks to minimize a function subject to a constraint. The function to be minimized is the departure from surface smoothness

$$\epsilon_s = (A+p-q)^2 + (B-p-q)^2 + (C-p+q)^2 + (D+p+q)^2$$
$$= 4(p-p_0)^2 + 4(q-q_0)^2 + \epsilon_0$$

If we were to plot contours of constant ϵ_s in gradient space we would obtain a set of concentric circles centered about the point (p_0, q_0) as in Figure 9a. Note that these contours are not the reflectance map $R(p, q)$ but a separate concept entirely. Now if we superimpose the contour of the reflectance map $R(p, q)=I(x, y)$ we obtain Figure 9b. It is evident that the point sought in gradient space is the point (p, q) nearest to (p_0, q_0) and on the contour $R(p, q)=I(x, y)$. Therefore, the line from (p_0, q_0) to the optimal (p, q) must be perpendicular to the contour if (p, q) is to be the nearest such point. Note that the vector $[R_p \ R_q]$ is perpendicular to the contour. Therefore, the vector $[(p-p_0) \ (q-q_0)]$ must be parallel to $[R_p \ R_q]$. Mathematically speaking

$$[(p-p_0) \ (q-q_0)] \times [R_p \ R_q] = 0 \tag{4.34}$$

or equivalently

$$(p-p_0)R_p - (q-q_0) R_q = 0$$

Thus

$$(p-p_0)R_p = (q-q_0) R_q \tag{4.35}$$

which accounts for Equation (4.32).
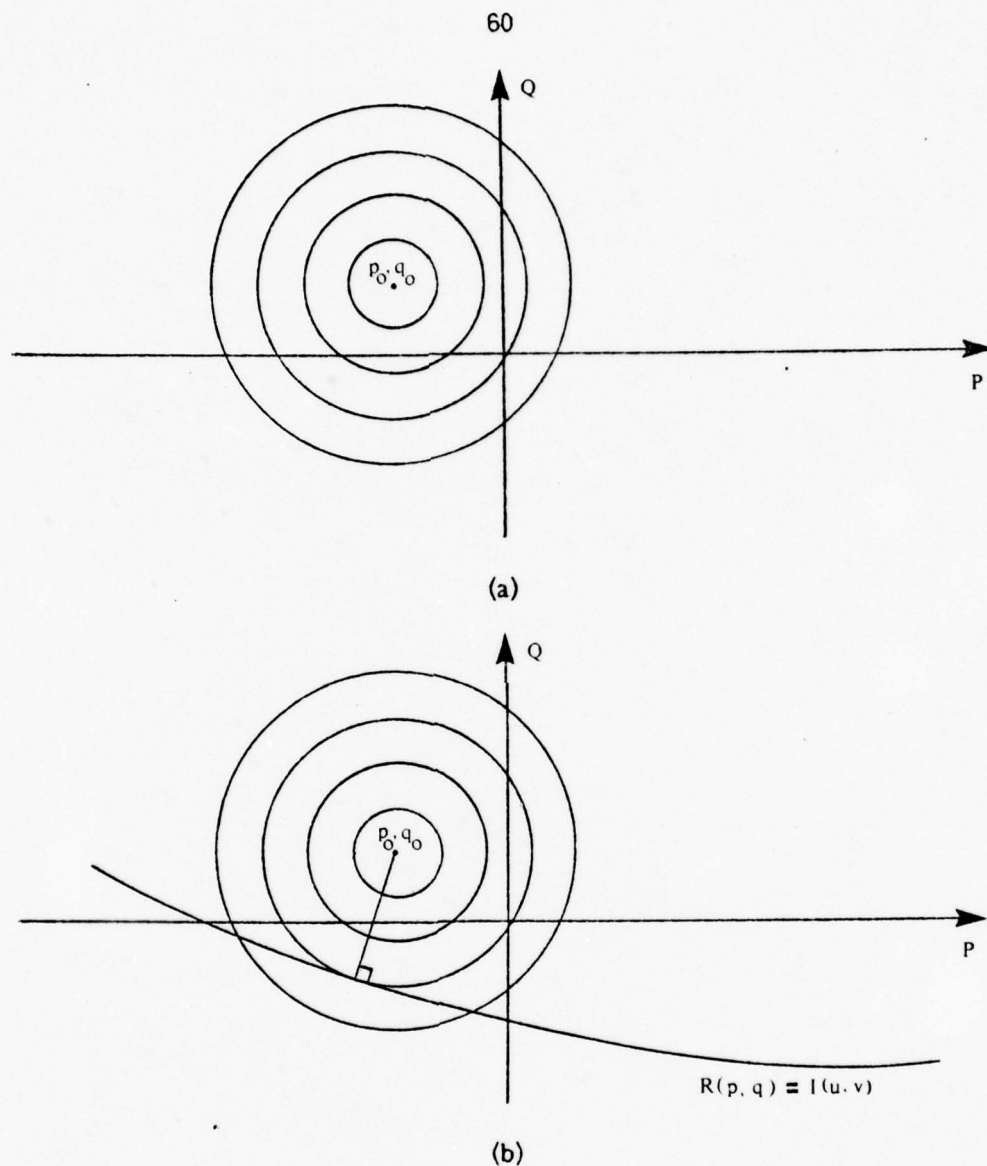
(a)

$$R(p, q) = I(u, v)$$

(b)

Figure 9 Lagrange Multipliers

Contours of constant ϵ_s are plotted in (a). The reflectance map contour satisfying the constraint $R(p, q) = I(x, y)$ is superimposed in (b). The desired gradient point (p, q) is that point on the reflectance map contour which has the lowest value of ϵ_s. This must occur where the reflectance map contour is tangent to a contour of constant ϵ_s.

4.4.2 Steepest Descent

Presented here is a rapidly convergent method for minimizing Equation (4.23) based on the algorithm of Fletcher and Powell [1963]. Their method is similar in theory to most other minimization routines but possesses two valuable properties.

It is fast compared to most other methods.

It submits to analysis of its stability and rate of convergence.

4.4.2.1 Geometric Interpretation

Before getting into the details of one particular method of minimization, it is enlightening to visualize the process. All conventional minimization methods use some sort of "hill sliding" strategy which incrementally approaches the absolute minimum.

The error function ϵ at a given image point and at a given stage in the computation is a single-valued function of p and q which can be plotted in gradient space. Figure 10 shows contours of constant error of a typical error function ϵ. The common method of steepest descent modifies the current assessment of the surface gradient as shown in the figure. Each step in gradient space is perpendicular to a contour of constant error. By iterating this procedure, one continually approaches the minimum. This simple method of steepest descent suffers from slow convergence rates for some functions. All the other minimization algorithms are principally the same but attempt to speed the convergence in various ways. The method of Fletcher and Powell is one example which is highly efficient and computationally simple.
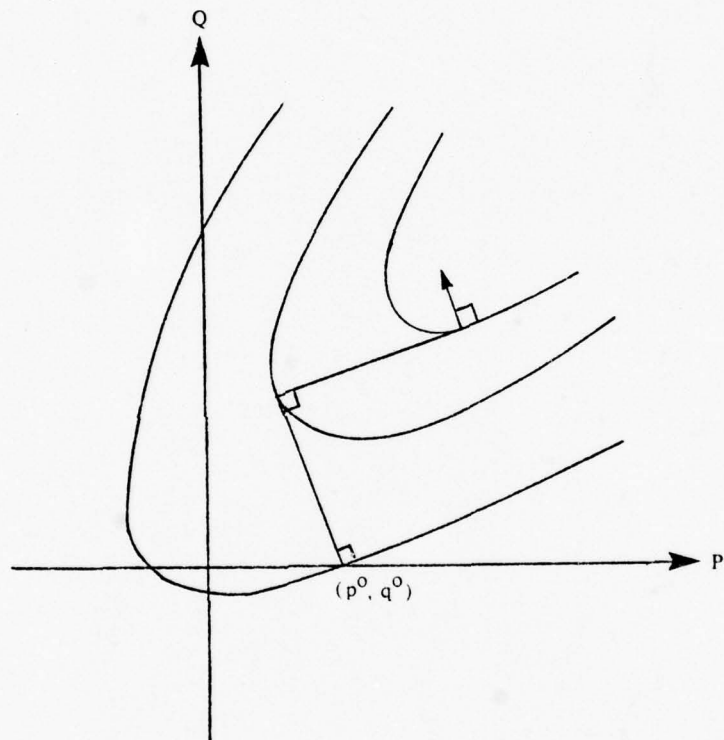
Figure 10 Minimization by Steepest Descent
Contours of constant error are shown. Suppose the current gradient point is (p^0, q^0). The next gradient is obtained by minimizing the function along the line through (p^0, q^0) perpendicular to the contour at that point. The minimum is approached by iterating this process. Each step is always perpendicular to the last and the process can be very slowly convergent.

4.4.2.2 The Method of Fletcher and Powell

This method makes use of the gradient with respect to p and q of our error function ϵ. Accordingly, it is applicable only in those cases where this gradient is defined. So if the reflectance map is not known analytically, this method cannot be used.

First some notation. $\epsilon(p, q)$ is of course the error function to be minimized. Define the vector $w=[p\ q]$ to be its arguments and $g=[\partial\epsilon/\partial p\ \ \partial\epsilon/\partial q]$ to be its gradient. If for the moment ϵ is assumed to be locally quadratic, then

$$\epsilon(w) = \epsilon_0 + \sum_i a_i w_i + \frac{1}{2}\sum_i \sum_j H_{ij} w_i w_j$$

or

$$= \epsilon_0 + aw + \frac{1}{2}w^T Hw \tag{4.36}$$

in matrix notation where ϵ_0 and a are constants and H is the Hessian matrix of the error function. Then

$$g = a + Hx \tag{4.37}$$

We can approximate the displacement between the point w and the actual minimum w_0 as

$$w_0 - w = -H^{-1}g \tag{4.38}$$

In Fletcher and Powell's method, the matrix H^{-1} is not evaluated directly; instead a matrix G is used which may initially be chosen to be any positive definite symmetric matrix. This matrix is modified after the ith iteration using the information gained by moving down the direction $s^i = -G^i g^i$ in accordance with Equation (4.38). The modification is such that σ^i, the step toward the minimum down the line $w^{i+1} = w^i + \lambda s^i$ is an eigenvector of the matrix $G^{i+1}H$. This ensures that as the procedure converges, G tends to H^{-1} evaluated at the

minimum w_0. It is convenient to use the identity matrix initially for G so that the first direction taken is down the line of steepest descent.

Let the current point be w^i with gradient g^i and matrix G^i. The iteration can then be stated as follows.

Set $s^i = -G^i g^i$ (4.39)

Obtain α^i such that $\epsilon(w^i + \alpha^i s^i)$ is a minimum with respect to λ along the line $w^i + \lambda s^i$ and $\alpha^i > 0$. α^i can always be chosen to be positive.

Set $\qquad\qquad\qquad \sigma^i = \alpha^i s^i$ (4.40)

Set $\qquad\qquad\qquad w^{i+1} = w^i + \sigma^i$ (4.41)

Evaluate $\epsilon(w^{i+1})$ and g^{i+1}. Note that g^{i+1} is orthogonal to σ^i. In other words, the line of steepest descent is perpendicular to the direction last moved so

$$\sigma^{iT} g^{i+1} = 0 \qquad (4.42)$$

Set $\qquad\qquad\qquad h^i = g^{i+1} - g^i$ (4.43)

Set $\qquad\qquad\qquad G^{i+1} = G^i + A^i + B^i$ (4.44)

where

$$A^i = \frac{\sigma^i \sigma^{iT}}{\sigma^{iT} h^i}$$

$$B^i = -\frac{G^i h^i h^{iT} G^i}{h^{iT} G^i h^i}$$

Set $i = i+1$ and repeat.

The method of obtaining the minimum along a line is not crucial to the algorithm. Fletcher and Powell suggest a procedure which uses cubic interpolation [Fletcher and Powell, 1963]. There are a great number of methods available for minimizing a function along a line and the reader is free to choose his favorite [Davidon, 1959].

4.4.3 The Gauss-Seidel Method

The next method to be explored starts out in a more direct way than either of the earlier two. The desire is to differentiate the error function ϵ (Equation 4.23) and solve for the gradient which simultaneously sets both derivatives to zero.

The partial derivatives of Equation (4.23) are easily found to be

$$\partial\epsilon/\partial p = 2(A-B-C+D) + 8p + 2\rho[R(p, q) - I(x, y)]R_p = 0$$
$$\partial\epsilon/\partial q = 2(-A-B+C+D) + 8q + 2\rho[R(p, q) - I(x, y)]R_q = 0$$

$$(4.45)$$

To solve these for p and q explicitly would require the reflectance map $R(p, q)$ and its derivatives to be known analytically. Furthermore the solution would be available explicitly only for certain simple reflectance functions.

Recall that after each iteration of relaxation, we will have p and q only approximately, because the parameters A, B, C and D are not really constants but change slightly after each iteration. Thus it is not important to be able to solve Equations (4.45) exactly. An approximation will do because the currently correct gradient will be different during the next iteration. The approximation is motivated as follows. Equations (4.45) can be rearranged as

$$p = \tfrac{1}{4}(-A+B+C-D) - \sigma[R(p, q) - I(x, y)]R_p$$
$$q = \tfrac{1}{4}(A+B-C-D) - \sigma[R(p, q) - I(x, y)]R_q$$

$$(4.46)$$

where $\sigma=\rho/4$.

The method of Gauss-Seidel [Froberg, 1969; Hamming, 1973] finds a solution to a system of equations by expressing them in the form

$$x_j^{i+1} = f(x_1^i, x_2^i, \ldots, x_n^i)$$

such that a new value of x_j can be calculated from the old values of x_i for all i. It is easy to see that Equations (4.46) are in this form.

$$p^{i+1} = \tfrac{1}{4}(-A^i + B^i + C^i - D^i) - \sigma[R(p^i, q^i) - I(x, y)]R_p\big|_{p^i, q^i}$$

$$q^{i+1} = \tfrac{1}{4}(A^i + B^i - C^i - D^i) - \sigma[R(p^i, q^i) - I(x, y)]R_q\big|_{p^i, q^i}$$

(4.47)

In words, these equations state that the new value of p is obtained from the old values of p and q and the old values of A, B, C and D which are functions of the old values of the neighboring p's and q's. By iterating this process at a particular image point, we can find the gradient which minimizes ϵ. Because we have to iterate all calculations for the relaxation scheme anyway, we need only perform one iteration of Equation (4.47) during each step of relaxation. The idea is that we can save time by having the local and global iterations converge simultaneously.

Three methods of finding the minimum of ϵ have been described. Other approximations, which might yield faster convergence rates are left to the reader's discretion and taste. The particular algorithm used to minimize the error function is not central to the theory of the numerical shape-from-shading algorithm. The best one is dependent upon the form of the function to be minimized, so the reflectance map partially determines the best method. The Gauss-Seidel Method (Equations 4.47) has been used most in the work reprted here because of its simplicity. In practice, the method to be used must be matched to the resources and application at hand.

An important restriction is eliminated by the Gauss-Seidel Method. All the other methods, including the analytic approach, require exact knowledge of R_p and R_q. Because

of the form of Equations (4.47), the Gauss-Seidel Method does not need these values exactly. They may be obtained approximately by interpolation of an empirical reflectance map and the algorithm will still converge to the exact solution. For the other methods, an error in determination of R_p or R_q may result in a corresponding error in determination of the gradient.

5. ANALYSIS OF THE ALGORITHM

The previous chapter described the numerical algorithm proposed for shape-from-shading. It would be very nice to be able to summarize its performance by, say, a single number but, as the reader may suspect, this is not possible. When comparing two shape-from-shading algorithms, there are many features, both quantitative and qualitative, that must be considered: What types of objects can be used? What kind and how many light sources? How many views? What initial conditions are needed? How accurate is the solution? How sensitive is the solution to noise in the image? To noise in the reflectance map? To the position of the light source? The purpose of this chapter is twofold: First it serves to clarify the functioning of the algorithm. Second, it answers some of the above questions. It is presented as a series of topics, each one analyzing a different aspect of the algorithm.

5.1 Examples

We give here a sampling of the performance of the algorithm. Two surfaces with different shapes and different photometric properties are presented and analyzed. Besides illustrating the concepts of the last chapter, they are offered as partial, preliminary evidence that the numerical algorithm does work as proposed.

5.1.1 The Lambertian Sphere

Some shape-from-shading algorithms have been applied to determining the shape of a sphere with a lambertian surface. For purposes of comparison, a lambertian sphere is analyzed here.

The image to be used as input appears in Figure 11. The image has been produced synthetically, assuming a lambertian surface with a single distant point source located at (0.7 0.3) in gradient space. An orthographic projection has been used.

A convenient form [Horn, 1977a; Marr, 1977; Stevens, 1979] for depicting the shape of a surface is what I shall call the *needle diagram*. It consists of line segments or *needles* positioned at various points in the image. The lengths of the needles represent the degree of slant at the various points on the object's surface; the orientations of the needles represent the directions of tilt (the directions of steepest descent). More specifically, the length of each needle is $\sqrt{p^2 + q^2}$ and the orientation is the projection of the local surface normal onto the plane of the image -- thus the needle diagram is a viewer-centered description of surface curvature. A needle diagram for the sphere of Figure 11 is given in Figure 12. We find this a convenient representation because it is exactly the representation found by the numerical shape-from-shading algorithm. We can use the needle diagram to summarize the current assessment of shape as determined by the algorithm after each stage of computation. Therefore, applying the algorithm to the image in Figure 11 should yield the needle diagram of Figure 12 exactly if it is correct.

For purposes of simplicity in implementing the algorithm, attention is restricted to a square inscribed in the circular projection of the sphere. Thus we will not attempt to
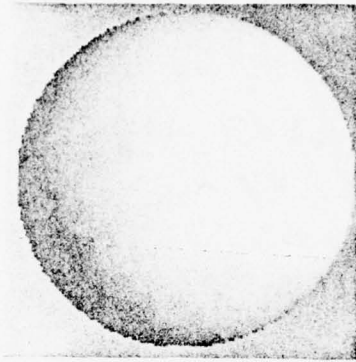
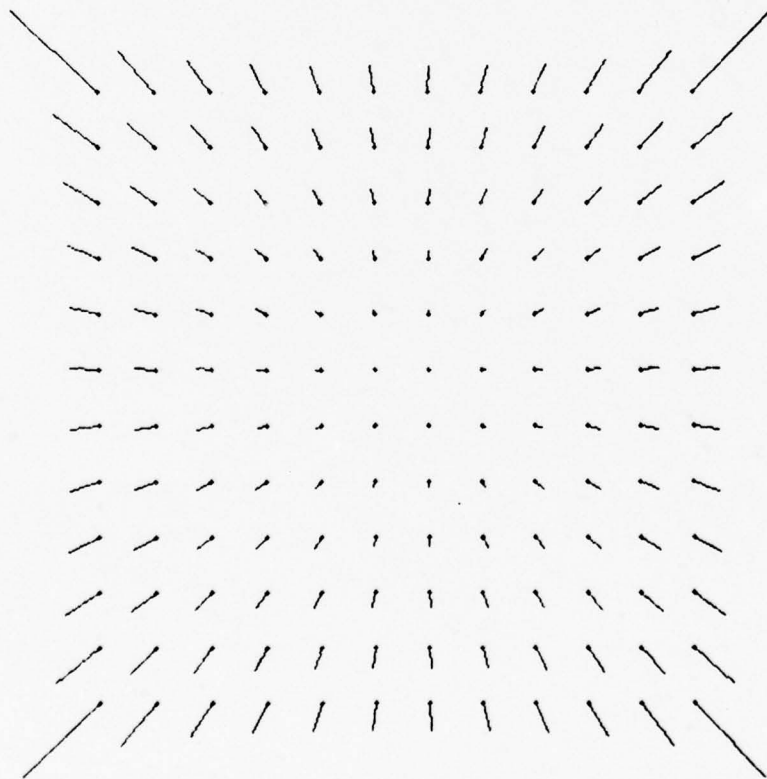Figure 11 Synthetic Image of a Lambertian Sphere

Figure 12 Needle Diagram of a Sphere
Each needle points in the direction of steepest descent on the surface.

determine the shape of the entire surface of the sphere but only a portion of the spherical surface instead. This subimage is selected to contain no self-shadowed portions of the sphere. It should be emphasized that the shape of the subimage in no way affects the operation of the algorithm, because approximately the same solution will be found regardless of the shape of the selected subimage in consideration. This property is essential to any shape-from-shading algorithm.

Initial values at every image point are chosen to be gradient (0, 0). More about initial values appears in Section 5.5. The gradients of the image points at the edge of the subimage are held fixed at the correct value throughout the computation. Therefore, Figure 13(a) is the needle diagram of the algorithm's initial guess at surface curvature. Subsequent diagrams show the convergence of the algorithm to the solution. Careful study of these diagrams demonstrates how a particular gradient is forced to a new value by its neighbors. The asymmetry exhibited is a by-product of the order of application of the local operators. For reasons discussed in Section 5.2, the operators are applied in a clockwise, square spiral toward the center. We have chosen to find the gradients at 100 points selected by a 10x10 square mesh superimposed on the image. Defining the average error in local surface orientation as the average over the 100 points of the angular difference between the assessed gradient and the true gradient, we find that the average error of the initial guess (Figure 13(a)) is nearly 30°. The first few iterations show how some gradients are quickly found correctly, others are temporarily opposite their final actual value, and some have not changed at all. By thirty iterations, (Figure 13(i)), nearly all gradients have been found accurately and the average error in local surface orientation has dropped to 3.1 degrees.

(a) Initial values

(b) 1 iteration

(c) 2 iterations
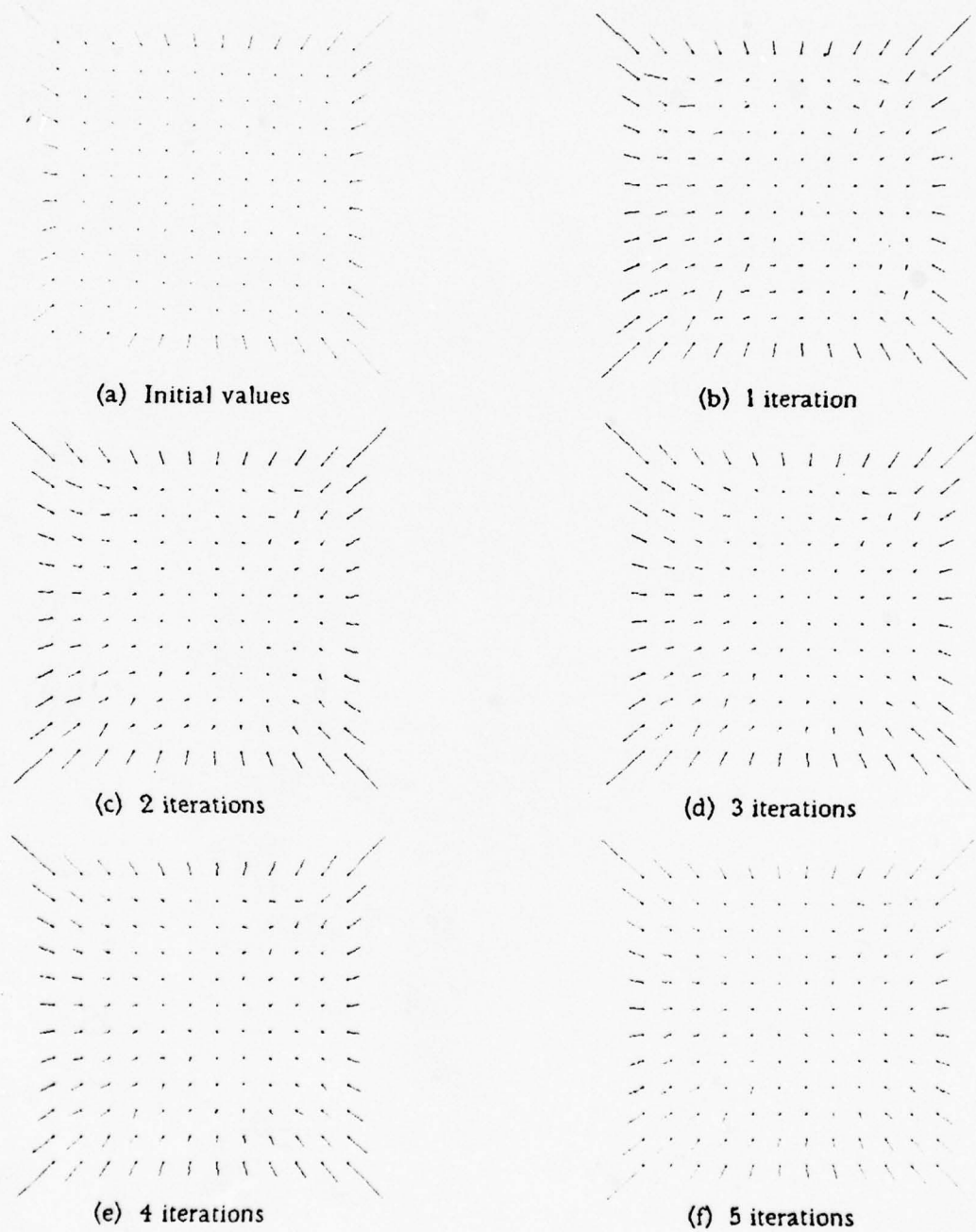
(d) 3 iterations

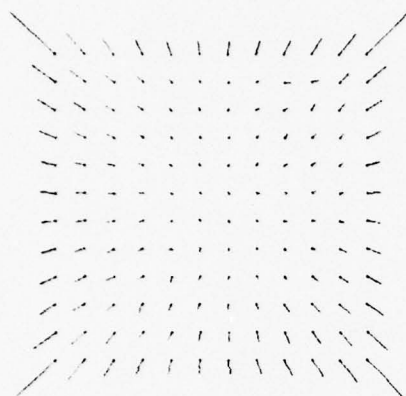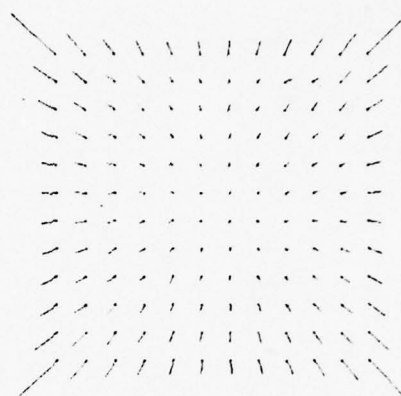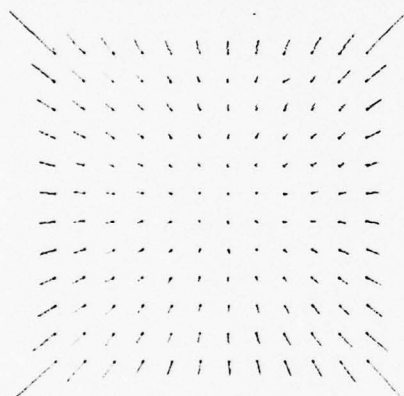(e) 4 iterations

(f) 5 iterations

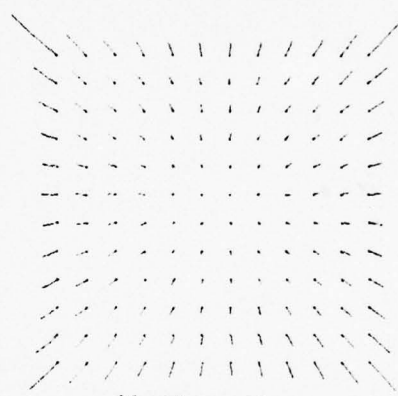Figure 13 Convergence of the Algorithm for the Lambertian Sphere
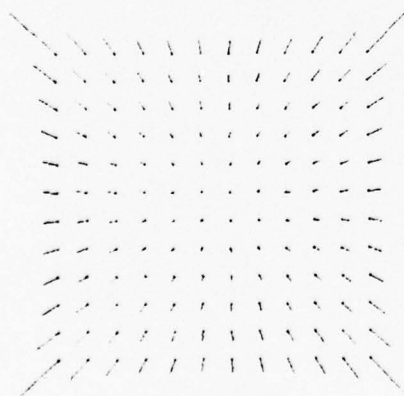
(g) 10 iterations

(h) 20 iterations

(i) 30 iterations

(j) 40 iterations

(k) 50 iterations

(l) Graph of average error
vs. iterations

Figure 13 (continued)

After fifty iterations, the algorithm is still converging and the average error is below two degrees. A plot of the average error in local surface orientation against the number of iterations (Figure 13(l)) shows the exponential character of the convergence. This plot is typical for the convergence for all surface shapes and reflectance maps.

5.1.2 A Lunar Waffle

In the last section, we analyzed an image of a surface with only one convexity. We will now study the performance of the algorithm on an image of a surface with many convexities and concavities. The image is shown in Figure 14. The equation of the surface $z=f(x, y)$ is

$$z = \sin(0.9x) + \sin(1.1y) \tag{5.1}$$

The shape of this surface is similar to a rectangular "waffle". For purposes of generating the synthetic image, the reflectance map $R(p, q)$ was chosen to be the same as the reflectance map of the material in the maria of the moon.

$$R(p, q) = 1 + 0.3p + 0.7q \tag{5.2}$$

The shape of a portion of the waffle is depicted in the needle diagram of Figure 15. We again start with initial values of $(p, q)=(0, 0)$ at all image points so that the initial estimate of surface shape is as shown in Figure 16(a). The remaining frames illustrate the convergence toward the true shape. As can be seen, the general shape of the surface has been determined within only five iterations. After fifty iterations, the average error in local surface orientation has dropped under 1.1° and continues to gradually decrease as predicted by the graph of Figure 16(l).
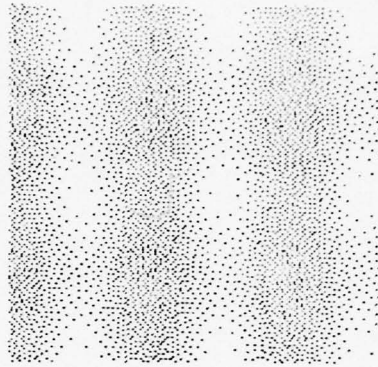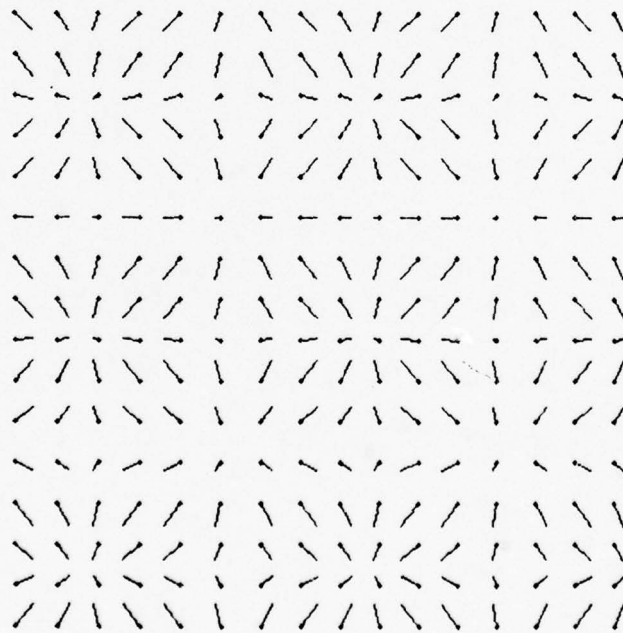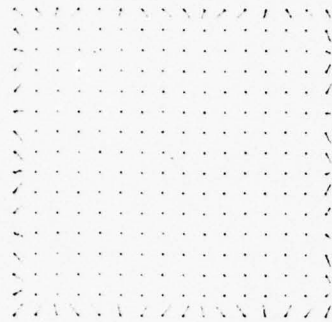
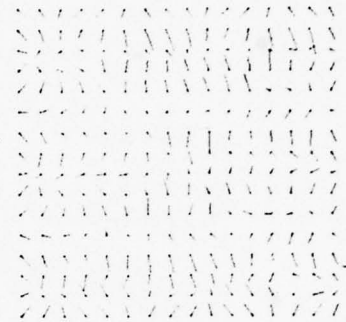Figure 14 Synthetic Image of a Lunar Waffle

Figure 15 Needle Diagram of the Waffle
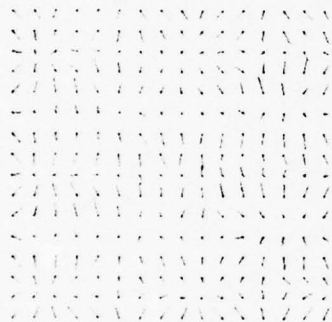Each needle points in the direction of steepest descent on the surface.
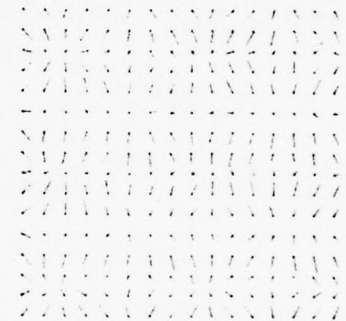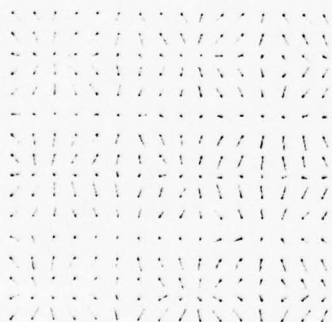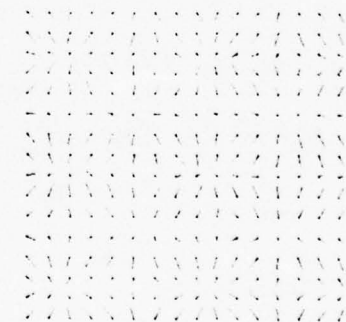
(a) Initial values

(b) 1 iteration
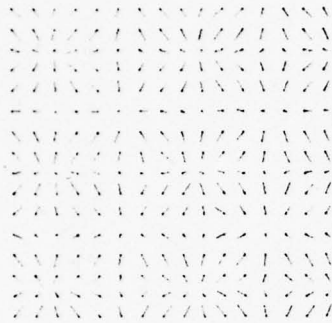
(c) 2 iterations

(d) 3 iterations

(e) 4 iterations
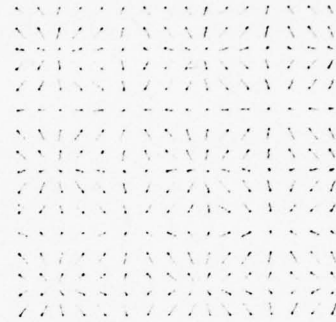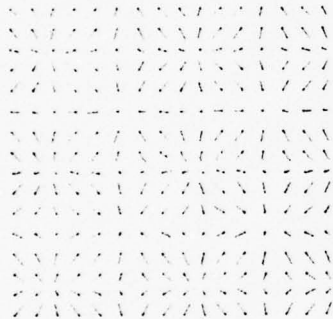
(f) 5 iterations

Figure 16 Convergence of the Algorithm for the Lunar Waffle
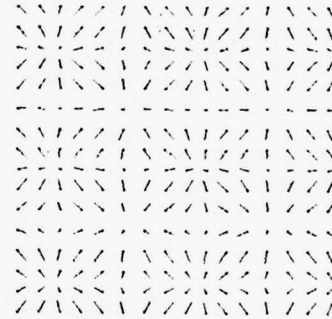
77

(g) 10 iterations

(h) 20 iterations

(i) 30 iterations
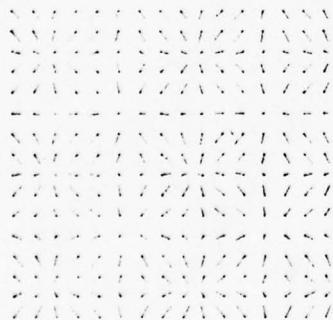
(j) 40 iterations

(k) 50 iterations

(l) Graph of average error
vs. iterations

Figure 16 (continued)

5.1.3 Terrain

When analyzing satellite photos, one is concerned with determining the shape of some complicated landform, not a simple mathematical function as in the preceding examples. Because the approximation of Equation (4.17) becomes more accurate as the grid resolution increases, a better estimate of shape is attained when the sampling resolution is great.

Synthetic images have been produced using Digital Terrain Models (DTM's), and analyzed by the algorithm. Experimentation has shown that the estimate of shape always converges toward the true topography, however it tends not to be found exactly. The relatively coarse resolution of DTM's prohibits Equation (4.17) from being exact, and the result is convergence to a somewhat inaccurate shape. For the DTM's studied, the topography was always correctly determined within $7°$ average error in local surface orientation and sometimes closer than $3°$ average error. It is enlightening to note that the error is spread throughout the surface points and is not the result of inaccurate determination of several critical points. When analyzing satellite photos, the resolution is determined by the resolution of the image which may be much better than the resolution of the available DTM's. Therefore much more accurate determination of surface topography may be expected, provided all other assumptions are valid, of course.

5.2 Stability

As mentioned earlier, there are two iterations converging simultaneously so the analysis of the stability of the numerical shape-from-shading algorithm is split into two parts. First, it will be shown that the minimization of the error term ϵ at each image point is stable, assuming the coefficients are constant. Second, the relaxation scheme is examined and shown to be stable when the minimization of ϵ is stable.

5.2.1 Stability of the Minimization Methods

In Section 4.4, three methods for solving Equation (4.23) were proposed. In the Lagrange Multiplier Method, stability is not a question because the minimum is isolated in one analytic step. The other two methods are iterative and must be shown to be stable. The method of Fletcher and Powell, while considerably more complex than the Gauss-Seidel Method, allows analytic analysis whereas Gauss-Seidel does not. For this reason a proof of the stability of the Fletcher-Powell method only is given. Empirical evidence is the only support offered for the stability of the Gauss-Seidel Method.

It is usual for descent methods to be stable because one ensures that the function to be minimized is decreased by each iteration. It will now be shown that the direction of search s^i defined in Equation (4.39) is downhill, so that α^i can always be chosen to be positive. Because g^i is the direction of steepest descent, the direction s^i will be downhill if and only if

$$-s^i g^i = g^{iT} G^i g^i \tag{5.3}$$

is positive. We want the direction of search s^i to be downhill for all possible g^i so we must

prove that G^i is positive definite for all i.

Assume G^i is positive definite and consequently that α^i is positive. It must be shown that for any vector x, $x^TG^{i+1}x \geq 0$. Because the square root of a positive definite matrix exists, we may define $\mu=\sqrt{G^i}\,x$ and $\nu=\sqrt{G^i}\,h^i$. From Equation (4.44) and the fact that G^i is symmetric, we have

$$x^TG^{i+1}x = x^TG^ix + \frac{x^T\sigma^i\sigma^{iT}x}{\sigma^{iT}h^i} - \frac{x^TG^ih^ih^{iT}G^ix}{h^{iT}G^ih^i}$$

$$= \frac{\mu^T\mu\ \nu^T\nu - (\mu^T\nu)^2}{\nu^T\nu} + \frac{(x^T\sigma^i)^2}{\sigma^{iT}h^i}$$

$$\geq \frac{(x^T\sigma^i)^2}{\sigma^{iT}h^i} \qquad\qquad (5.4)$$

by making use of Schwartz's inequality.

Now

$$
\begin{aligned}
\sigma^{iT}h^i &= \sigma^{iT}g^{i+1} - \sigma^{iT}g^i && \text{from (4.43)} \\
&= -\sigma^{iT}g^i && \text{from (4.42)} \\
&= -\alpha s^{iT}g^i && \text{from (4.40)} \\
&= \alpha^i g^{iT}G^ig^i && \text{from (4.39)} \\
&\geq 0 && \qquad (5.5)
\end{aligned}
$$

By induction then, $x^TG^{i+1}x>0$ for all non-trivial vectors x and hence G^{i+1} is positive definite.

As stated in the beginning of the proof, the direction of search will always be downhill if G^i is positive definite for all i. Therefore, each iteration of the search reduces the error and the process is stable for each image point (u, v) [Fletcher and Powell, 1963].

5.2.2 Stability of the Relaxation Scheme

It has just been shown that the Fletcher-Powell Method of minimizing the error ϵ at each image point (u, v) is stable. We now turn attention to the analysis of the stability of the entire relaxation scheme. To this end we apply the von Neumann criterion for the stability of finite difference approximations [Richtmeyer and Morton, 1967].

> *Definition*: Examine all solutions to determine whether any of them increase without limit even when their initial values are bounded. If any of them do, the scheme is said to be *unstable*. Otherwise, it is *stable*.

Since the relaxation scheme is the top-level of the shape-from-shading algorithm, the algorithm will be stable if each iteration of the relaxation is stable.

Consider allowing the algorithm to run for i complete iterations and then abruptly halting it. At every image point (u, v) there exists a current assessment of the gradient (p_{uv}^i, q_{uv}^i). We will now single out the gradient at a generic image point (u, v) and note how its value affects the values of its neighbors. Consider the following trick. Take the frozen computation and update the value of the gradient at point (u, v) only. That is, apply the local operator only to image point (u, v) to obtain p_{uv}^{i+1} and q_{uv}^{i+1}. Using the circumflex to denote values of p and q obtained after iteration i in the *modified* problem we define

$$\hat{p}_{uv}^i = p_{uv}^{i+1} \quad \text{and} \quad \hat{q}_{uv}^i = q_{uv}^{i+1} \tag{5.6}$$

Further define

$$\pi = p_{uv}^{i+1} - p_{uv}^i \quad \text{and} \quad \theta = q_{uv}^{i+1} - q_{uv}^i \tag{5.7}$$

That is, π is the amount by which p_{uv}^i is changed to obtain $\hat{p}_{uv}^i$ and θ is the amount by which q_{uv}^i is changed to obtain $\hat{q}_{uv}^i$. We will now press the restart button and continue to

use the circumflex to denote values of p and q obtained in this modified problem.

The template in Figure 8 can help us visualize the computation. First consider the template centered over the point $(u+1, v+1)$. We see from Equation (4.21) that the value of $\hat{p}^{i+1}_{(u+1)(v+1)}$ is found to be

$$\hat{p}^{i+1}_{(u+1)(v+1)} = p^{i+1}_{(u+1)(v+1)} + \pi/4 - \theta/4 \tag{5.8}$$

Similarly

$$\hat{q}^{i+1}_{(u+1)(v+1)} = q^{i+1}_{(u+1)(v+1)} + \theta/4 - \pi/4 \tag{5.9}$$

By centering the templates over the other neighbors of (u, v) we find

$$\hat{p}^{i+1}_{(u+1)(v-1)} = p^{i+1}_{(u+1)(v-1)} + \pi/4 + \theta/4 \qquad \hat{q}^{i+1}_{(u+1)(v-1)} = q^{i+1}_{(u+1)(v-1)} + \theta/4 + \pi/4$$
$$\hat{p}^{i+1}_{(u-1)(v-1)} = p^{i+1}_{(u-1)(v-1)} + \pi/4 - \theta/4 \qquad \hat{q}^{i+1}_{(u-1)(v-1)} = q^{i+1}_{(u-1)(v-1)} + \theta/4 - \pi/4$$
$$\hat{p}^{i+1}_{(u-1)(v+1)} = p^{i+1}_{(u-1)(v+1)} + \pi/4 + \theta/4 \qquad \hat{q}^{i+1}_{(u-1)(v+1)} = q^{i+1}_{(u-1)(v+1)} + \theta/4 + \pi/4$$
$$\hat{p}^{i+1}_{u(v+1)} = p^{i+1}_{u(v+1)} + \pi/2 \qquad \hat{q}^{i+1}_{u(v+1)} = q^{i+1}_{u(v+1)} + \theta/2$$
$$\hat{p}^{i+1}_{u(v-1)} = p^{i+1}_{u(v-1)} + \pi/2 \qquad \hat{q}^{i+1}_{u(v-1)} = q^{i+1}_{u(v-1)} + \theta/2$$
$$\hat{p}^{i+1}_{(u+1)v} = p^{i+1}_{(u+1)v} - \pi/2 \qquad \hat{q}^{i+1}_{(u+1)v} = q^{i+1}_{(u+1)v} - \theta/2$$
$$\hat{p}^{i+1}_{(u-1)v} = p^{i+1}_{(u-1)v} - \pi/2 \qquad \hat{q}^{i+1}_{(u-1)v} = q^{i+1}_{(u-1)v} - \theta/2$$

Now consider image point (u, v). We have

$$\hat{p}^{i+1}_{uv} = p^{i+2}_{uv} \qquad \hat{q}^{i+1}_{uv} = q^{i+2}_{uv} \tag{5.10}$$

Furthermore, all other image points (that is, points further than one grid point away from (u, v)) are unchanged:

$$\hat{p}_{jk} = p_{jk} \qquad \text{and} \qquad \hat{q}_{jk} = q_{jk}$$

$$\forall j \notin \{u-1, u, u+1\} \qquad \forall k \notin \{v-1, v, v+1\}$$

We now see that the change in p or q at any image point X due to the change at another image point Y is smaller in magnitude than the increase at Y that caused it, after one iteration. Furthermore, Equation (5.10) guarantees the stability of point Y, because the local

operators have been shown to be stable. Therefore, an increase in p or q at Y produces bounded increases in p or q at all other image points. Since the increase at every image point is stable and the effect of each increase is bounded, each iteration is stable according to the von Neumann criterion. By induction then, the entire relaxation scheme is stable because each iteration is stable. The algorithm can only be unstable when the local operators are unstable.

5.3 Convergence

It is desirable to have the algorithm converge as rapidly as possible. It is clear by now that there are actually two iterations happening at the same time. The determination of the gradient (p, q) which minimizes the error function ϵ is approximated by successive iterations using one of the minimization schemes presented in Section 4.4. Additionally, the propagation of constraints afforded by the local smoothing operator in the relaxation scheme represents an approximation by successive iterations. It is intuitive then (although not proven) that optimization of the overall convergence rate is attained by separate optimizations of the convergence rates of both of the iterative components.

First we note that whenever the algorithm converges, it converges to the correct solution. Proofs of the convergence of the minimization schemes are not included here. The Lagrange Multiplier Method is a one-step analytic solution and requires no proof. The proof of the Fletcher-Powell Method is given in [Fletcher and Powell, 1963]. We can conclude that the error function ϵ can be minimized. Whenever the shape-from-shading problem is "well-formed", it will be possible to minimize ϵ to be arbitrarily close to zero.

Recalling Equation (4.7)

$$\epsilon = \epsilon_s + \rho \, \epsilon_r$$

we see that both ϵ_s and ϵ_r must approach zero as well. Restricting ϵ_s to zero implies that the derived surface is a real, smooth surface. Further restricting ϵ_r to zero implies that the derived realizable surface will generate the same image as the one used to determine the shape. This means that whenever the mapping from image to surface is unique, the algorithm will determine that surface after some number of iterations given that all its assumptions are valid. Therefore, the algorithm converges to the correct solution whenever that solution exists.

The number of iterations required to reach the correct solution within a given tolerance depends on the convergence rate of the minimization method used. Thus, an optimal minimization method will be optimal as a local operator for the numerical shape from shading algorithm. The actual convergence rate realized will be slower than the convergence rate of the minimization method because of the hysteresis of the relaxation scheme. Since one application of the local operator is used n^2 times in one iteration on an n x n image, it is highly desirable to select a minimization method which is simple to compute as well as rapidly convergent.

5.4 Boundary Values

The regions for which we wish to perform shape-from-shading are bounded by the size of the image or some subsection of the image that is of interest. The local operators that we have described are not applicable to image points which do not have all eight neighbors

present. The easy solution is to surround the region with the actual values of the gradients at each of the border points. Then the local operators will never extend off the image to an undefined point. The examples shown in this thesis have all focused on a square region surrounded by a square of actual gradients as was shown in Figure 13.

An alternative solution is based on the definition of the error function ϵ_s which depends on the gradients of all eight neighbors. Given the square tessellation that we use for our grid, there are four possible types of image points.

(1) Interior points: These points have all eight neighbors defined and the method of Equation (4.19) can be used as always.

(2) Edge points: These points lie on an edge of the region. We can construct an expression for ϵ_s based on two loops instead of the usual four. For the case of a point on the left side of the image we get
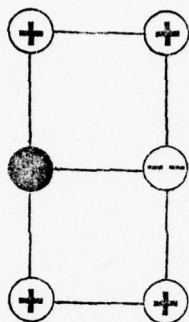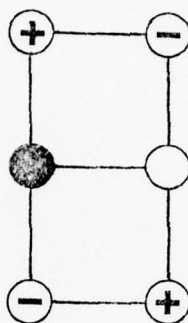
$$\epsilon_s = (A+p-q)^2 + (B-p-q)^2$$

Minimizing this equation with respect to p and q results in the templates of Figure 17 which gives a new value for the gradient based on the gradients of the five defined neighbors of the edge point (u, v). A similar result can be derived for each of the other three possible sides of a region.

(3) Corner points: These are points which lie in a convex corner of the region. In this case, only one loop is available for computation of ϵ_s. For the lower-left corner we find

$$\epsilon_s = (A+p-q)^2$$

The templates for the optimal p and q of a corner point are given in Figure 18 and are found to depend on the three defined neighbors of the corner point.

(4) Concave corner points: These points occur only if there are concavities in the region of interest. They permit three loops to be used for calculating ϵ_s. The development is completely analogous to the previous cases and is omitted here. An expression can be found based on the gradients of its seven defined neighbors using three loops.

$$p_{uv}^{i+1}$$

$$p_{uv}^i \qquad\qquad q_{uv}^i$$

$$p_{uv}^{i+1} = \tfrac{1}{2}(B - A)$$

$$= \tfrac{1}{2}\left[q^i_{(u+1)(v-1)} - q^i_{u(v-1)} - q^i_{(u+1)(v+1)} + q^i_{u(v+1)} + p^i_{u(v-1)} + p^i_{(u+1)(v-1)} - 2p^i_{(u+1)v} + p^i_{(u+1)(v+1)} + p^i_{u(v+1)}\right]$$
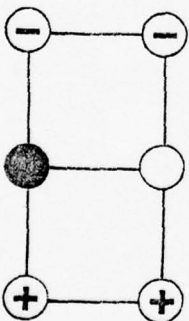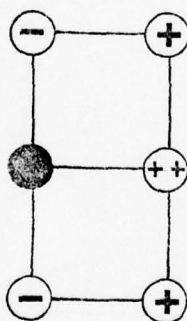
$$q_{uv}^{i+1}$$

$$p_{uv}^i \qquad\qquad q_{uv}^i$$

$$q_{uv}^{i+1} = \tfrac{1}{2}(A + B)$$

$$= \tfrac{1}{2}\left[p^i_{u(v-1)} + p^i_{(u+1)(v-1)} - p^i_{(u+1)(v+1)} - p^i_{u(v+1)} + q^i_{(u+1)(v-1)} - q^i_{u(v-1)} + 2q^i_{(u+1)v} + q^i_{(u+1)(v+1)} - q^i_{u(v+1)}\right]$$

Figure 17 Templates for Edge Points
Templates of the local smoothness operators are shown here for image points on the left edge of the region. Templates for the other three edges are similar in form.

$$p_{uv}^{i+1}$$
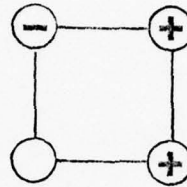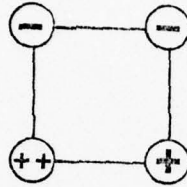
$$p_{uv}^i \qquad\qquad q_{uv}^i$$

$$p_{uv}^{i+1} = p_{uv}^i + \tfrac{1}{2} A$$

$$= \tfrac{1}{2} [2p_{uv}^i + p_{(u+1)v} + q_{(u+1)v} + q_{(u+1)(v+1)} - p_{(u+1)(v+1)} - p_{u(v+1)} - q_{u(v+1)}]$$

$$q_{uv}^{i+1}$$
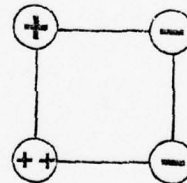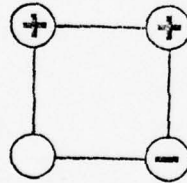
$$p_{uv}^i \qquad\qquad q_{uv}^i$$

$$q_{uv}^{i+1} = q_{uv}^i - \tfrac{1}{2} A$$

$$= \tfrac{1}{2} [2q_{uv}^i - p_{(u+1)v} - q_{(u+1)v} - q_{(u+1)(v+1)} + p_{(u+1)(v+1)} + p_{u(v+1)} + q_{u(v+1)}]$$

Figure 18 Templates for Corner Points

Templates of the local smoothness operators are shown here for the image point in the lower left-hand corner of the region. Templates for the other three corners are similar in form.

The numerical shape-from-shading algorithm is an implementation of a solution to the differential equation (3.1) due to Horn. There we found the necessity of an initial curve which intersects all the characteristics. We still need that initial curve in the numerical approach. It can be manifested implicitly in the boundary values or given as a curve of points within the region. The use of a square boundary of actual gradients actually overdetermines the system. This is not a problem because the relaxation scheme is inherently prepared to resolve such conflicts. If the overdetermination is grossly incorrect, then spurious results will be obtained. An underdetermined system (due to lack of an initial curve or appropriate boundary values) will prevent convergence to the true solution.

5.5 Initial Values

At any given point in the computation, the algorithm has a current assessment of surface shape. Future assessments of shape are calculated from the image, the reflectance map, and the current assessment. In order to begin the computation, there must be an initial assessment of the gradient at every image point. We call these *initial values* and they are simply some assignment of shape from which to begin the calculation.

In theory, the choice of the initial values should be arbitrary. One would hope that the algorithm would determine the same surface shape regardless of the choice of initial values. In practice, this is found to be true. For the sake of uniformity, the initial value of (0, 0) has generally been assigned as the gradient for each image point. This assignment corresponds to a flat plane perpendicular to the viewing direction.

While the choice of initial values doesn't affect the solution, it can affect the time

required to reach that solution. Of course, one would like to start as close to the true shape as possible. A flat planar surface, being some kind of average of all possible surfaces, is likely to produce a short convergence time. For this reason, it has usually been used as the initial value of surface shape in the course of this research.

5.6 Errors in Boundary Values

What happens if the boundary values specified are not correct? In practice, one would not expect to be able to determine the boundary values exactly. Furthermore, there is the chance that some of the values will be specified incorrectly. The hope is that the algorithm will not fail completely if this is the case. However, the nature of the algorithm might permit errors to be propagated throughout the region, thereby nullifying its usefulness.

We return to the image of the lunar waffle to approach this problem. To study the effect of an incorrect boundary value, the boundary value at image point (0, 5) was set to gradient (0, 0). This represented a deviation of $53°$ from its true orientation of (0.900, 0.983). After fifty iterations, the surface shape represented by the needle diagram of Figure 19 was reached. We see that the shape was calculated fairly accurately for most points. We can see the effect of the incorrect boundary value more clearly by studying the difference between the shape of Figure 19 and the real shape as was shown in Figure 15. If one actually subtracts the needles of Figure 19 from the true needles using simple vector subtraction, the diagram of Figure 20 is obtained. From it, it is easily seen that the error is only propagated about three image points from the incorrect value. This result is found to be true in general and is
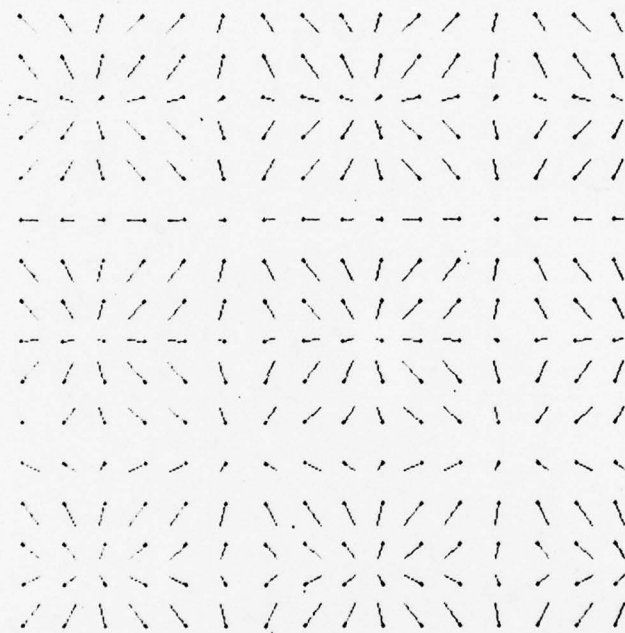
Figure 19 An Incorrect Boundary Value
The gradient of the boundary point (0, 5) was incorrectly specified as (0, 0) before running the algorithm. This needle diagram shows the surface shape determined by the algorithm after fifty iterations. Compare this diagram with the true shape in Figure 15.
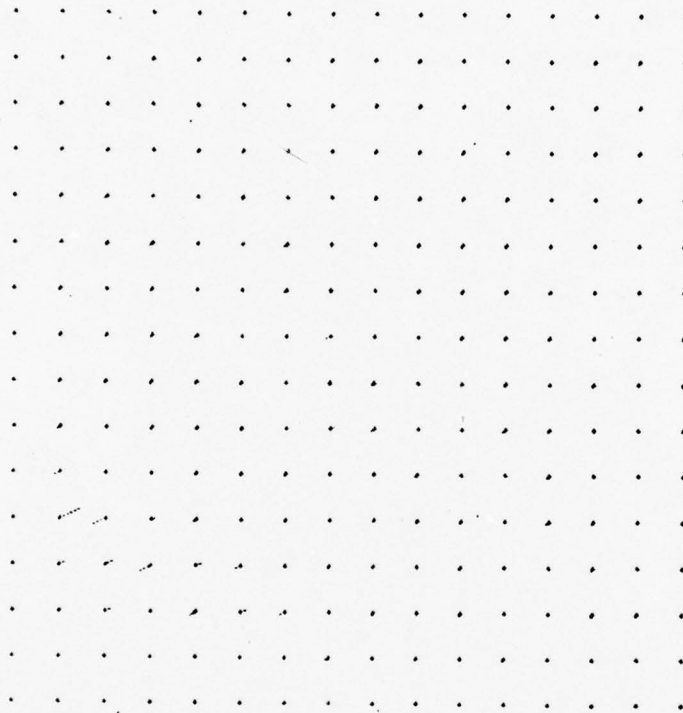
Figure 20 The Difference Diagram for an Incorrect Boundary Value
This diagram was obtained by subtracting the true surface shape from the shape computed after running the algorithm for fifty iterations starting with the Figure 19 as the initial values. Note that very little error is propagated more than three image points.

a desirable property to have. When one considers images of 500x500 points or larger, an incorrect boundary value is seen to have very little effect indeed.

One may be concerned over the combined effects of many boundary errors. The algorithm is surprisingly resistant to combined errors and will be pretty close unless there are so many errors systematically in the same direction, that the required initial conditions are not well specified. It would be meaningless to calculate the precise amount of error propagation because it will depend on a particular boundary error, surface shape, reflectance map and resolution. A rule of thumb is that the maximum propagation of a single error is limited to a radius of three grid points.

5.7 Noise in the Image

The examples shown in this thesis have been computed from synthetic images. For a practical application, one must face up to the deficiencies of the real world. Here we focus on the issue of incorrect intensity values $I(x, y)$ recorded in an image.

If $I(x, y)$ is not what it is supposed to be as predicted by our model of image formation, the algorithm will be fooled and an incorrect assessment of shape will be obtained. In fact, this is precisely the reason why facial cosmetics are used [Horn, 1970]. By skillfully applying make-up to the face, a person can deceive an observer into perceiving the shape of the face as different from its actual shape. In our terms, the application of makeup corresponds to changing the photometric properties in certain regions. Thus our assumption of constant reflectance map at all image points has been violated and the algorithm will be fooled as the human was. Other causes for an image intensity to be different from our

model's prediction are an aberration in the film, noise and quantization error in the digitized image.

To study the consequences of an incorrect intensity value, the intensity value at image point (8, 7) in the image of the waffle was purposely changed from 0.95 to 0.0, a 50% change in the total range of intensities in the image. Figure 21 shows the solution obtained after fifty iterations. At first glance, the modification of intensity at point (4, 5) appears to have had limited effect on the solution. Again, the needle diagram of the difference provides some insight. Figure 22 is obtained by subtracting Figure 21 from the true shape of Figure 15. Again we find that very little error is propagated more than three grid points from the modified value.

This implies that for real applications, a few bad intensity values won't destroy the solution. If this were not true, the algorithm would be useless. In large images, the algorithm will be fooled by defects or cosmetics locally, but the effect will not propagate to other portions of the image.

Another deficiency of the real-world is created by our representation of images. When an image is digitized, an intensity value is represented as one of a finite number of grey-levels. Thus the intensity is not known exactly, but only within some degree of precision. Experimentation has shown that surface shape can be reliably recovered using images with as few as sixteen grey-levels. The exact precision required is dependent upon the reflectance map and the true shape of the surface. When working with digitized images, one finds that the average error in local surface orientation does not converge to zero but levels off somewhere between zero and eight degrees depending upon the number of grey-
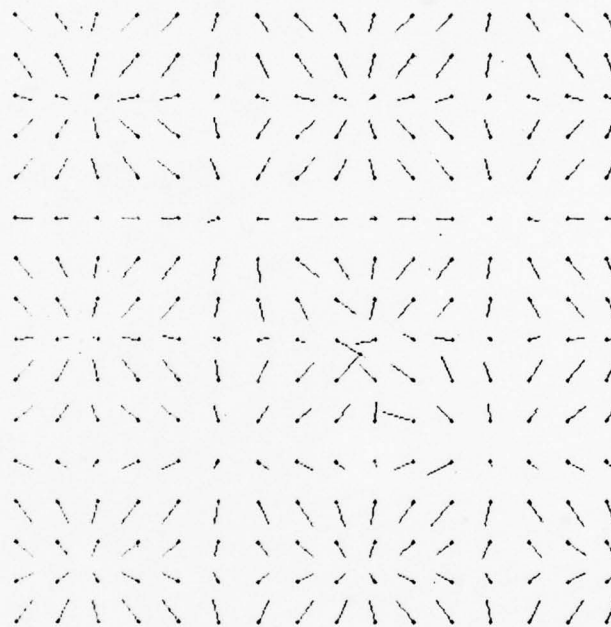
Figure 21 An Incorrect Intensity Value

This diagram was obtained after running the algorithm for 50 iterations on an image which had one incorrect intensity value (at point (8, 7)).
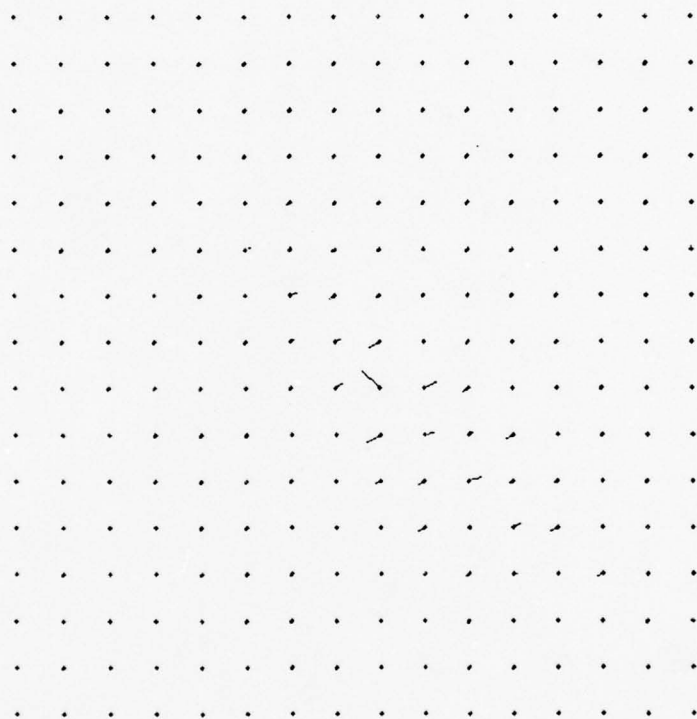
Figure 22 The Difference Diagram for an Incorrect Intensity Value
Subtracting Figure 21 from the true shape of Figure 15 yields this diagram. Note that very little error has propagated more than three image points from the position of the incorrect intensity value at (8, 7).

levels.

Besides large individual errors, we might expect all the image intensities to be off slightly. This may be due to flaws in the imaging process or noise in the image. Again, it is impossible to predict the effects precisely, but as another rule of thumb, the algorithm can withstand errors of up to 10% of the total magnitude of intensity in the image at all image intensities at once.

5.8 Inaccurate Determination of the Reflectance Map

Yet another real-world source of potentially dangerous error is the accuracy of the determination of the reflectance map. All the empirical methods discussed in Section 2.3.2 are subject to measurement error. Furthermore, the analytic functions are only approximations to the physical reality. Thus in general, it is impossible to determine the reflectance map exactly. Again we must ask what implications this has for the numerical shape-from-shading algorithm.

Experimentation has shown that satisfactory results are obtained even when the reflectance map is not accurate. The exact tolerance of the algorithm is difficult to measure because of the complexity of comparing two reflectance maps, but a general trend is evident. In all cases, small errors in the reflectance map produce small errors in the resultant estimate of shape. The form of this error is manifested in the failure of the algorithm to converge to the exact shape. Instead, a plateau is reached in which the average error in determination of the local surface normal no longer decreases. Furthermore, there exists a threshold of error in the assumed reflectance map, beyond which the "solution" bears little resemblance to

END

DATE
FILMED

3--79

DDC

the actual shape. Exactly where this threshold lies is difficult to ascertain. To be a little more quantitative, several experiments were performed in which the light source for the assumed reflectance map was "moved" from the true light source position used in computing the image, thus changing the reflectance map. The actual relationship between the magnitude of the light source's movement and the ultimate convergence level is highly dependent upon the reflectance function used and to a lesser degree on several other factors. In the example of a square region of a spherical lambertian surface, the light source was "moved" 6.4° from its position when the image was calculated. Even so, the algorithm was able to determine local surface orientations to within 5° of the truth. Apparently the lunar reflectance map is more forgiving, allowing determination to within 2° for the same movement of light source. The thresholds of maximum movement are not sharply defined, but spread out between 2° and 20° for most reflectance functions.

5.9 Dependence of the Convergence Rate

We are naturally concerned with the number of iterations required to achieve a given accuracy in the assessment of surface shape. Experimentation has shown that this number (call it the *convergence rate*) is highly dependent on many factors. The sampling resolution, the shape of the surface, the initial estimate of that surface, the value of ρ in our error function, and the form of the reflectance map all come into play. Furthermore, they are not independent factors but have mutually dependent effects on the convergence rate.

The sampling resolution is determined by the number of grid points that we select from an image and not its absolute size. Thus we can speak of an n x n image where we

wish to determine the gradients at n x n points. It might be feared that the convergence rate increases with n^2, the number of points to be determined. Fortunately, the relaxation scheme propagates information in all directions simultaneously and the convergence rate is found to increase only with n. Experimentation has borne this out while a mathematical proof is given by [Garabedian, 1967].

The number of iterations required to reach a given accuracy in assessment of surface shape. depends on the initial assessment. The closer one starts, the better the convergence rate. If one starts with a flat planar surface, then one would expect the convergence rate to depend only on the difference between these surfaces, given all other factors held constant. In general, this is not true. The actual convergence rate depends on the actual shape of the surface in some as of yet inexplicable way.

Recalling Equation (4.7), we wonder if the choice of ρ affects the convergence rate. The answer is a definite yes. Given all other factors held constant, there exists a finite positive value of ρ which is optimal. Values smaller than this generally produce slower convergence rates and values "near" zero do not produce convergence at all. Values of ρ much greater than the optimal value either do not converge or produce unstable results.

The shape of the reflectance map is also a factor in the convergence rate. For example, lambertian surfaces generally converge slower than "lunar" surfaces of the same shape. The quality of a reflectance map that determines its effect on the convergence rate is unknown. For a surface with given photometric properties, the position of the light source affects the convergence rate as well. We can specify the position of a distant light source as (ϕ, θ), where ϕ is the inclination of the source above the horizon (the elevation) and θ is its

rotation from North (the azimuth). It has been found that smaller source elevations produce faster convergence rates. This fact cannot be exploited, however, because smaller source elevations generally produce larger shadow areas. One would not expect the azimuth of the source to play much of a role, but in fact, it does. Even images of surfaces which are circularly symmetric show convergence rates which depend on the azimuth because of the sequential application of the local operators. The shape of the reflectance map is a very important and complicated factor in determining the convergence rate.

One surprising factor is the order of application of the local operators. In fact, if the local operators are applied in parallel, the sequence may not converge at all. To see this, think of the simplified case of determining a single-valued function by averaging the values of the four nearest neighbors at each iteration. Applying these operators in parallel produces what I call the *checkerboard effect*. After some number of iterations, all red squares have one value and all black squares have another. Each subsequent iteration finds all red squares taking on the values of the black squares and black squares taking on the last red square values. The current assessment is seen to flip back and forth in this manner unendingly. The checkerboard effect is exhibited by our complex local operators but in a more subtle form. The flipping can be eliminated by scanning serially, as one reads a book. This type of scanning requires n iterations for information to propagate from the bottom to the top of an n x n image. The time can be cut in half by scanning in a "square spiral". The corresponding reduction in the convergence rate using the spiral scan is small -- about 10% fewer iterations than required by the linear, book scan.

5.10 Varying Reflectance Maps

Throughout this thesis, the surface has been assumed to be isotropic. It will now be shown that this restriction can be relaxed slightly.

First consider the horrendous situation where every object point has different photometric properties. The reflectance map would be a function of the *position* of a surface point as well as its orientation. The simple image forming equation becomes

$$I(x, y) = R(x, y, p, q) \tag{5.11}$$

If the reflectance map were somehow known for every point in the image, then the algorithm can simply substitute each reflectance map into Equation (5.11) when analyzing the corresponding image point and the correct description of surface shape will be found. This is a result of the fact that reflectance is a strictly local property (when the effects of shadows and mutual illumination have been ignored). Unfortunately, one would not expect to be able to determine the reflectance map $R(p, q)$ for every point in the image in practice. One example of when this approach might be reasonable concerns Landsat images [Short *et. al.*, 1976]. If one knows the surface cover beforehand and is attempting to determine the surface topography, then the reflectance function for a wheat field can be used where a wheat field is known to exist; the reflectance function for Kentucky blue grass can be used in a field of Kentucky blue grass; etc. Note that determining which image points correspond to what types of surface cover may be a difficult and time-consuming task. Additionally, one would expect the reflectance function of a growing field to change with the seasons. Thus determining the reflectance function for an area of known surface cover would not be so simple.

As a second case, suppose the reflectance function at each image point is an unknown multiplicative factor of the reflectance function of every other point. That is,

$$R(x, y, p, q) = \rho(x, y) R(p, q) \tag{5.12}$$

where $\rho(x, y)$ represents the surface albedo. This situation might arise when one considers an objected covered with varying colors of the same type of paint. It might also be an approximation to the reflectance map of objects composed of several different materials. Without knowing the albedo $\rho(x, y)$ at each image point, no shape-from-shading algorithm could determine the surface shape. More information is necessary to counterbalance the additional degree of freedom afforded by $\rho(x, y)$.

A second image of the same scene obtained from the same view angle but different light source position can supply the needed information [Horn, Woodham & Silver, 1978]. With two images we have

$$
\begin{aligned}
I_1(x, y) &= \rho(x, y) R_1(p, q) \\
I_2(x, y) &= \rho(x, y) R_2(p, q)
\end{aligned}
\tag{5.13}
$$

There is no problem registering these two images because they were obtained with the same viewer object geometry. The unknown albedo at each image point can be eliminated by forming the quotient of the two images, obtaining

$$\frac{I_1(x, y)}{I_2(x, y)} = \frac{\rho(x, y) R_1(p, q)}{\rho(x, y) R_2(p, q)} = \frac{R_1(p, q)}{R_2(p, q)} \tag{5.14}$$

Defining $I_{12}(x, y) = I_1(x, y)/I_2(x, y)$ and $R_{12}(p, q) = R_1(p, q)/R_2(p, q)$ we have

$$I_{12}(x, y) = R_{12}(p, q) \tag{5.15}$$

which is now in a form for applying the numerical shape-from-shading algorithm using the

derived function $R_{12}(p, q)$ as the reflectance map.

One can go one step further to recover the surface albedo $\rho(x, y)$ at each point as well. Construct a synthetic image using $R_1(p, q)$ as reflectance map and with the same viewer-object geometry as $I_1(x, y)$, based on the determination of surface shape obtained from the shape-from-shading algorithm. Call this synthetic image $S_1(x, y)$. Then it is immediately apparent from Equation (5.13) that

$$\rho(x, y) = \frac{I_1(x, y)}{S_1(x, y)} \tag{5.16}$$

Therefore, if the change in surface reflectance can be modeled as a multiplicative factor of the reflectance map at each object point (*i.e.* the only thing that varies is the albedo), then two images are sufficient to recover both surface topography and surface albedo at each point in the images.

As an aside, it is interesting to study the shape of the derived "reflectance map" $R_{12}(p, q)$. When R_1 and R_2 are lambertian, we find that contours of $R_{12}(p, q)$ are a family of straight lines all intersecting at a single point -- the intersection of the terminators (shadow lines) of each reflectance map. (See Figure 23.) Curiously enough, the quotient of two images is unrecognizable to the human eye, but the numerical shape-from-shading algorithm can recover the surface shape precisely using the strange "reflectance map" $R_{12}(p, q)$ (which, incidentally, is physically impossible for any real surface).

A point source near the object provides a third interesting case. Now the phase angle G is no longer constant even when the viewer is distant, so the reflectance map $R(p, q)$ varies over the image. Nevertheless, a solution is possible if the reflectance function is

known analytically as $\phi(i, e, g)$. Here $\phi(i, e, g)$ can be expressed as $R(p, q, g)$ using Equations (2.5 - 2.7). Then the reflectance map will be known at all image points simply by calculating g at each point. In an implementation, each local operator will be a different, but defined, function, and the shape can be computed.
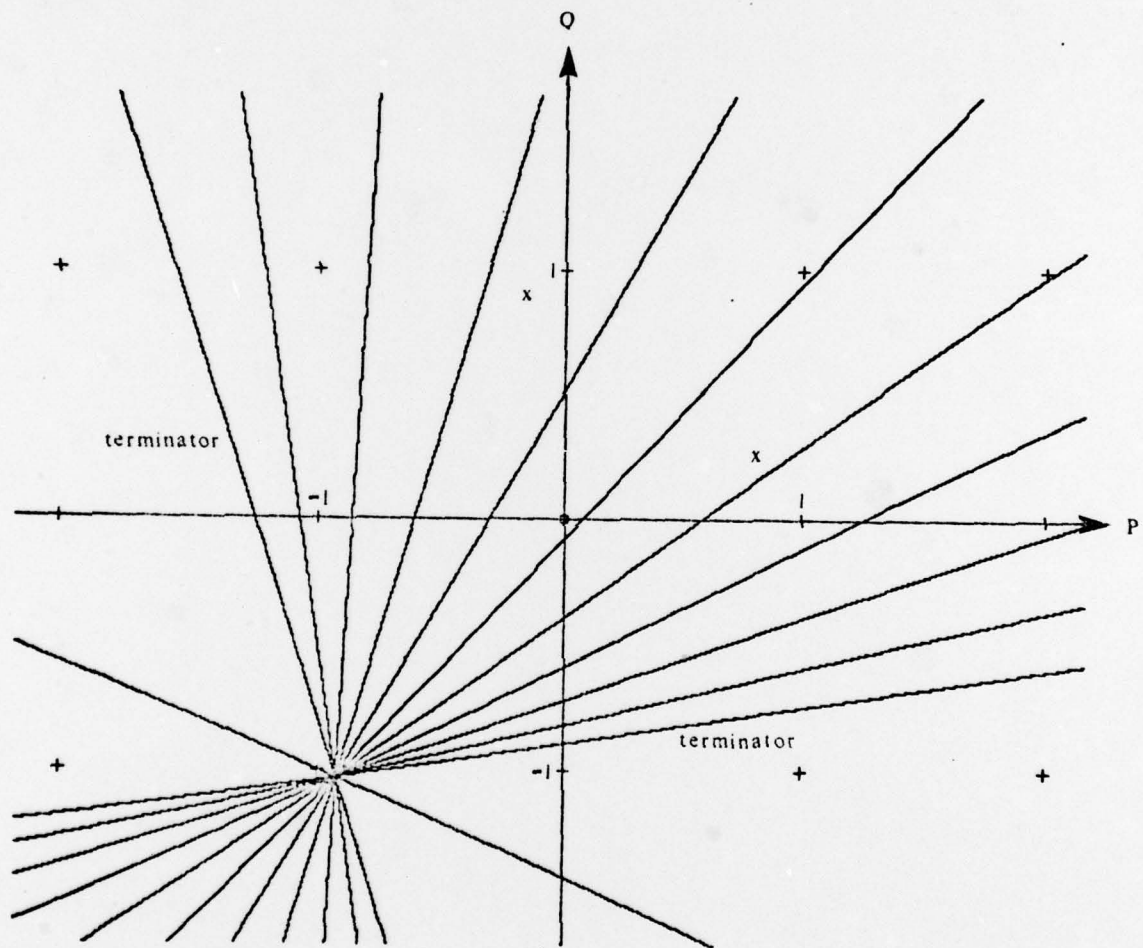

Figure 23 Contours of the Quotient of Two Lambertian Reflectance Maps

6. CONCLUDING REMARKS

6.1 What Remains To Be Done

While the numerical shape-from-shading algorithm (SFS) is not *the* universal shape-from-shading algorithm, it is a very capable method of performing shape-from-shading in a variety of applications. One can foresee more powerful methods in the future based to a greater or lesser extent on some of the notions exploited by SFS, but for the present, we may look at ways of improving the performance and applicability of this algorithm. The sections that follow outline some major renovations to SFS that would actually make it more powerful and which are not merely special purpose tricks.

6.1.1 Accelerating the Convergence Rate

It has been shown how the convergence rate can be improved by optimizing each component separately. This optimization is not straightforward. Many algorithms have been proposed for minimizing a function during the past century and the issue has still not been settled. Optimization of the relaxation scheme has not enjoyed as much attention. One may look to the methods of computer science to improve the convergence rate of the relaxation scheme. Some possibilities include various predictor-corrector methods of iteration and the use of larger templates to provide for faster propagation of information throughout the image. Perhaps an hierarchy of local operators would be useful. The optimal value of p is seen to change with each iteration. Sequential adjustment of p might

improve the convergence rate as well.

6.1.2 Effects of Shadows

It is well-known that shadows are of two varieties, namely cast shadows and self-shadows. For our purposes, a shadow is a region in an image with intensity equal to zero. Thus, no information about the shape of the surface within that region is available. Currently, SFS will try to force the gradients of points in the shadowed region to lie on the contour of $R(p, q)=0$. This is not valid and needs to be dealt with. One would hope that the presence of a shadow would not prevent SFS from determining topography from non-shadowed areas. It should be possible to work around shadowed areas by marking them and using the boundary operators of Section 5.4 on their edges.

Additionally, shadows do provide information about the shape of other parts of the surface and SFS is not prepared to deal with this. One may incorporate knowledge from these clues in novel ways to create a more applicable algorithm. Stevens [1979] provides some details about how shadows might be used to determine shape.

6.1.3 Effects of Mutual Illumination

When light is reflected from one part of a surface onto others, the reflectance map is altered at those points. Without knowledge of this, SFS will be fooled into identifying an incorrect surface shape. An impractical way to handle this is to specify the correct reflectance map in areas of mutual illumination. In this way, the knowledge of mutual illumination can be utilized by the algorithm.

An alternate method is based on the fact that the shape and photometric properties of an object completely determine the location and extent of all mutual illumination. Since SFS is designed to calculate shape, it ought to be capable of using its knowledge of shape to refine its knowledge of the reflectance map in areas of mutual illumination. One can envision a third layer of iterative approximation added to SFS to deal with this. As the surface shape is determined in one part of the image, the location and extent of mutual illumination in other parts of the image is also determined. This can be used to more accurately approximate the shape in those regions of mutual illumination. Hopefully, as the computation converges, the equation of image formation becomes satisfied, the assessment of topography becomes more continuous and the estimate of mutual illumination comes closer and closer to the actual mutual illumination.

6.1.4 Coping with Discontinuities

The numerical shape-from-shading algorithm possesses no capability of handling surface discontinuities. Such discontinuities must be predetermined and the algorithm must be applied in regions not containing any discontinuities. In practice, this becomes a large drawback. Machine parts typically contain many discontinuities and mountains obscure other mountains at low view angles. It might be possible to equip SFS with some sort of discontinuity detector because SFS is designed to determine shape. Once a discontinuity is located, the boundary operators of Section 5.4 can once again be used to protect information from crossing the discontinuity. Further use of discontinuity clues and occluding bounds is available in several recent works [Marr, 1976a; Stevens, 1979].

6.2 Relation to Other Work

6.2.1 Biological Systems

The numerical shape-from-shading algorithm was developed from a foundation of understanding the imaging process. We have studied how the physical world constrains the transformation from object space to image space, and this enabled us to theorize ways to invert the process. At no point in the research did knowledge or hypotheses about visual systems of animals affect the development of the algorithm or the underlying theory. For this reason, we would expect any resemblance between the numerical shape-from-shading algorithm and existing biological visual systems to be extremely unlikely. In retrospect, we find that any resemblance is indeed minimal. For example, the algorithm has been shown to require more time for increased resolution or increased accuracy. No analogous result has been demonstrated in animals.

6.2.2 Computer Systems

The numerical shape-from-shading algorithm was motivated by two concepts. The first was the partial differential equations for determining shape from a single image. The second was the constraint imposed by the imaging equation as in the photometric stereo approach. The result was the development of an iterative relaxation scheme which solves the partial differential equations numerically.

Aside from an analytic solution to those differential equations, the relaxation

scheme is the only available method to determine shape from a *single* image. Because analytic solutions are not normally available, the relaxation scheme becomes more important. Its simplicity and robustness make it a very practical scheme for real-world application. One can conceive of a hardware implementation which performs the relaxation using analog signals [Horn, 1974]. The speed available by such a device could offset the large number of iterations sometimes required.

It is interesting to note that the method of this paper and the binocular stereo approach can be integrated into a single system for performing image analysis. It turns out that the two methods are complementary -- one succeeds where the other stumbles. For example binocular stereo works best on images of scenes with many discontinuities while the numerical algorithm can (must) determine shape in regions containing no discontinuities. Similarly binocular stereo thrives on images of nonisotropic surfaces while the numerical algorithm succeeds on (requires) images of surfaces with constant photometric properties.

As noted in the introduction, image analysis can be divided into many subproblems of which the shape-from-shading problem is one. A discussion of how a shape-from-shading scheme might be incorporated into a comprehensive image analysis system is given by [Barrow & Tenenbaum, 1979].

7. REFERENCES

Allen, D. N. de G. (1954), *Relaxation Methods*, New York, McGraw-Hill, 1954.

Ames, W. F. (1977), *Numerical Methods for Partial Differential Equations*, Academic Press, New York, 1977.

Ammar, M. H. (1978), "A Real-Time System for Determining Reflectance Maps", S. B. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., May 1978.

Barrow, H. G. and Tenenbaum, J. M. (1978), "Recovering Intrinsic Scene Characteristics from Images", in *Computer Vision Systems*, Hanson and Riseman (eds.), 1979.

Booth, A. D. (1957), *Numerical Methods*, Butterworths, London, 1957.

Clowes, M. B. (1971), "On Seeing Things", in *Artificial Intelligence*, Vol. 2, pp. 79-112, 1971.

Curry, H.D. (1944), "The Method of Steepest Descent for Non-linear Minimization Problems", *Qu. App. Maths.*, Vol 2., p. 258.

Davidon, W. C. (1959), "Variable Metric Method for Minimization", A.E.C. Research and Development Report, ANL-5990 (Rev.).

Deist, F. H. and Sefor, L. (1967), "Solution of Systems of Non-linear Equations by Parameter Variation", *The Computer Journal*, Vol. 10., No. 1, May 1967.

Fletcher, R. and Powell, M. J. D. (1963), "A Rapidly Convergent Descent Method for Minimization", *The Computer Journal*, Vol. 6, pp. 163-168, 1963.

Freuder, E. C. (1976), "A Computer System for Visual Recognition Using Active Knowledge", M.I.T. Artificial Intelligence Technical Report 345, June 1976.

Froberg, C-E., (1969), *Introduction to Numerical Analysis*, Addison-Wesley Publishing Company, Reading, Mass., 1969.

Garabedian, P. R. (1967), *Partial Differential Equations*, John Wiley & Sons, Inc. New York, 1967.

Grasselli, A. (ed.) (1969), *Automatic Interpretation and Classification of Images*,
Academic Press, New York, 1969.

Guzman, A. (1968), "Computer Recognition of Three-Dimensional Objects in a Visual
Scene", MAC TR-59 (Thesis), Project MAC, M.I.T., December 1968.

Hamming, R. W. (1973), *Numerical Methods for Scientists and Engineers*, New York,
McGraw-Hill, 1973.

Horn, B. K. P., (1970), "Shape from Shading: A Method for Obtaining the Shape
of a Smooth Object from One View", M.I.T. Project MAC, Technical
Report 79, November 1970.

Horn, B. K. P. (1973) "The Binford-Horn Line Finder", M.I.T. Artificial Intelligence
Memo 285, December, 1973

Horn, B. K. P. (1974), "On Lightness", in *Computer Graphics and Information
Processing*, Vol. 3, pp. 277-299, December 1974.

Horn, B. K. P. (1975), "Obtaining Shape from Shading Information", in *The
Psychology of Computer Vision*, P.H. Winston (ed.), McGraw-Hill,
pp. 115-155, 1975.

Horn, B. K. P. (1977a), "Understanding Image Intensities," *Artificial Intelligence*,
Vol. 8, pp. 201-231, April 1977.

Horn, B. K. P. (1977b), "Using Synthetic Images to Register Real Images with Surface
Models," M.I.T. Artificial Intelligence Memo 437, August 1977.

Horn, B. K. P., Woodham, Robert J. and Silver, William M. (1978), "Determining
Shape and Reflectance Using Multiple Images", M.I.T. Artificial
Intelligence Memo 490, August 1978.

Horn, B. K. P. and Sjoberg, Robert, (1978), "Calculating the Reflectance Map",
M.I.T. Artificial Intelligence Memo 498, October 1978.

Householder, A. S. (1953), *Principles of Numerical Analysis*, McGraw-Hill,
New York, 1953.

Huffman, D. A. (1971), "Impossible Objects as Nonsense Sentences", in *Machine
Intelligence 6*, R. Meltzer and D. Michie (ed.), Edinburgh University Press,
pp. 295-323, 1971.

International Business Forms Industry (1969), *Optical Character Recognition and the Years Ahead*, The Business Press, Elmhurst, Ill., 1969.

Levenberg, K. (1944), "A Method for the Solution of Certain Non-linear Problems in Least Squares", *Qu. App. Maths*, Vol. 2, p. 164.

Lozano-Perez, T. (1977), "Parsing Intensity Profiles", *Computer Graphics and Image Processing*, Vol. 6, No. 1, February 1977.

Mackworth, A. K. (1973), "Interpreting Pictures of Polyhedral Scenes", in *Artificial Intelligence*, Vol. 4, pp. 121-137, 1973.

Marr, D. and Poggio, T. (1976), "Cooperative Computation of Stereo Disparity", M.I.T. Artificial Intelligence Memo 364, June 1976.

Marr, D. (1976a) "Analysis of Occluding Contour", M.I.T. Artificial Intelligence Memo 372, October 1976.

Marr, D. (1976), "Early Processing of Visual Information", *Phil. Trans. Roy. Soc. B*, pp. 483-524, 1976.

Marr, D. and Poggio, T. (1977), "A Theory of Human Stereo Vision", M.I.T. Artificial Intelligence Memo No. 451, November 1977.

Marr, D. (1977), "Representing Visual Information", M.I.T. Artificial Intelligence Memo 415, May 1977.

Nicodemus, F. E., Richmond, J. C. and Hisa, J. J. (1977), "Geometrical Considerations and Nomenclature for Reflectance", NBS Monograph 160, National Bureau of Standards, Washington, D.C., October 1977.

Nitzan, D., Brian, A. E. and Duda, R. O. (1977), "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis", *Proc. IEEE*, pp. 206-220, February, 1977.

Phong, B. T. (1975), "Illumination for Computer Generated Pictures", *CACM* Vol. 18, pp. 311-317, 1975.

Powell, M. J. D. (1962), "An Iterative Method for Finding Stationery Values of a Function of Several Variables", *The Computer Journal*, Vol 5., p. 147.

Pun, L. (1969), *Introduction to Optimization Practice*, John Wiley & Sons, Inc., New York, 1969.

Richtmeyer, R. D. and Morton, K. W. (1967), *Difference Methods for Initial-Value Problems*, Interscience Publishers, New York, 1967.

Roberts, L. G. (1965), "Machine Perception of Three-Dimensional Solids", in *Optical and Electro-optical Information Processing*, J. T. Tippett *et. al.* (ed.), M.I.T. Press, pp. 159-197, 1965.

Rosenbrock, H. H. (1960), "An Automatic Method for Finding the Greatest or Least Value of a Function", *The Computer Journal*, Vol. 3, p. 175, 1960.

Shah, B. V., Buehler, R. J., and Kempthorne, O. (1961), "The Method of Parallel Tangents (PARTAN) for Finding an Optimum", Office of Naval Research Report, NR-042-207 (No. 2).

Shirai, Y. (1975), "Analyzing Intensity Arrays Using Knowledge About Scenes", in *The Psychology of Computer Vision*, P. H. Winston (ed.), McGraw-Hill, pp. 93-113, 1975.

Short, N. M., Lowman, P. D., Freden, S. C., and Finch, W. A., (1976), *Mission to Earth: Landsat Views the World*, Scientific and Technical Information Office, National Aeronautics and Space Administration, Washington, D.C., 1976.

Stevens, K. A. (1979), "Surface Perception from Local Analysis of Texture and Contour", Ph.D. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., January 1979.

Strat, T. M. (1978), "Shaded Perspective Images of Terrain", M.I.T. Artificial Intelligence Memo 463, March 1978.

Szego, G. P. (1972), *Minimization Algorithms*, Academic Press, Inc., New York, 1972.

Turner, K. (1971), "Object Recognition Tests on the Mark 1.5 Robot", Edinburgh University School of Artificial Intelligence, MIP-R 92, December 1971.

Waltz, D. (1975), "Understanding Line Drawings of Scenes with Shadows", in *The Psychology of Computer Vision*, P. H. Winston (ed.), McGraw-Hill, pp. 19-91, 1975.

Wilde, D. U. (1966), "Program Analysis by Digital Computer", M.I.T. Project MAC Technical Report 43, 1966.

Winston, P. H. (1973), "The M.I.T. Robot", in *Machine Intelligence 7*,
B. Meltzer and D. Michie (ed.), Edinburgh University Press, pp. 431-463, 1973.

Winston, P. H. (1975), "Learning Structural Descriptions from Examples", in
The Psychology of Computer Vision, P. H. Winston (ed.), McGraw-Hill,
pp. 431-463, 1975.

Woodham, R. J. (1977), "A Cooperative Algorithm for Determining Surface Orientation
from a Single View," *Proc. 5th International Joint Conference on
Artificial Intelligence*, M.I.T., Cambridge, Mass., August 1977, pp. 635-641.

Woodham, R. J. (1978a), "Reflectance Map Techniques for Analyzing Surface Defects
in Metal Castings", M.I.T Artificial Intelligence Technical Report 457,
June 1978.

Woodham, R. J. (1978b), "Photometric Stereo", M.I.T. Artificial Intelligence Memo
No. 479, June 1978.

Young, I. T. (1969), "Automated Leukocyte Recognition", PhD Thesis, M.I.T., 1969.